

UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



INGENIERÍA INFORMÁTICA

PROYECTO FIN DE CARRERA

**“CREACIÓN DE UN ENTORNO 3D PARA
LA SIMULACIÓN DE TRÁFICO
URBANO”**

**Autor: Víctor Romero Pérez.
Tutor: Raúl Arrabales Moreno.**

Julio 2009





AGRADECIMIENTOS

Gracias a mi familia y amigos, por su apoyo y cariño que me han mostrado durante todos estos años de estudio.

Gracias a mis compañeros, por cada minuto que han compartido conmigo, disfrutando, aprendiendo y viviendo junto a mi, momentos irrepetibles que estarán siempre en mi memoria.

Agradecer a mi tutor D. Raúl Arrabales Moreno, por toda la ayuda prestada así como su total disponibilidad para poder realizar este proyecto.





INDICE

1. INTRODUCCIÓN:	8
1.1. ESCENARIO DEL PROYECTO.	8
1.2. OBJETIVOS.	9
1.3. GESTIÓN DEL PROYECTO.	11
1.4. ESTRUCTURA DE LA MEMORIA.	13
2. ¿POR QUÉ SKETCHUP?	14
2.1. INTRODUCCIÓN A LAS HERRAMIENTAS DE MODELADO 3D.	15
2.2. ANÁLISIS COMPARATIVO DE LAS DISTINTAS HERRAMIENTAS DE MODELADO 3D.	15
2.2.1. Autodesk 3ds Max 9 SP2:	17
2.2.2. Autodesk Maya:	18
2.2.3. Blender:	18
2.2.4. SolidWorks:	19
2.2.5. Sketchup:	20
2.2.6. Softimage Mod Tool:	20
2.2.7. Softimage XSI:	21
2.2.8. AC3D:	21
2.2.9. Solid Edge:	21
2.3. CONCLUSIONES SOBRE EL ANÁLISIS Y EL PORQUÉ DE ELEGIR SKETCHUP.	22
3. HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS.	23
3.1. Visual Studio 2008:	23
3.2. Robotics Developer Studio:	25
3.3. Netbeans:	26
3.4. Java.	27
3.5. C#.	28
3.6. Google Earth.	29
3.7. Sketchup.	30
4. ANALISIS DE REQUISITOS Y DISEÑO DEL SOFTWARE.	31
4.1. INTRODUCCION.	31
4.1.1. Alcance del software.	32
4.1.2. Visión General	32
4.2. DESCRIPCIÓN GENERAL.	33
4.2.1. Perspectiva del producto	33
4.2.2. Capacidades generales	33
4.2.3. Restricciones generales	34
4.2.4. Características de los usuarios	34
4.2.5. Entorno operacional	34
4.2.6. Suposiciones y dependencias.	35
4.3. DEFINICIÓN DE REQUISITOS.	35
4.4. IDENTIFICACIÓN DE REQUISITOS.	36
4.5. REQUISITOS DE USUARIO.	38
4.5.1. Requisitos de capacidad	38



4.5.2.	Requisitos de restricción	40
4.6.	CASOS DE USO.	42
4.6.1.	Diagrama de Casos de Uso General.....	42
4.6.2.	CU001 Caso de Uso Cargar Fichero Principal.	43
4.6.3.	CU002 Caso de Uso Editar Personas.	43
4.6.4.	CU003 Caso de Uso Editar Señales.	44
4.6.5.	CU004 Caso de Uso Editar Edificios.....	45
4.6.6.	CU005 Caso de Uso Editar Climatología.	46
4.6.7.	CU006 Caso de Uso Editar Mobiliario.	47
4.6.8.	CU007 Caso de Uso Almacenar Entorno Seleccionado.	47
4.6.9.	CU008 Caso de Uso Mostrar Manual de Usuario.	48
4.7.	DIAGRAMAS DE SECUENCIA.....	49
4.7.1.	Diagrama de secuencia DS001 para el caso de uso CU001.....	50
4.7.2.	Diagrama de secuencia DS002 para el caso de uso CU002.....	51
4.7.3.	Diagrama de secuencia DS003 para el caso de uso CU003.....	52
4.7.4.	Diagrama de secuencia DS004 para el caso de uso CU004.....	53
4.7.5.	Diagrama de secuencia DS005 para el caso de uso CU005.....	54
4.7.6.	Diagrama de secuencia DS006 para el caso de uso CU006.....	55
4.7.7.	Diagrama de secuencia DS007 para el caso de uso CU007.....	56
4.7.8.	Diagrama de secuencia DS008 para el caso de uso CU008.....	57
4.8.	TRAZABILIDAD.	58
4.8.1.	Matriz de trazabilidad.	59
4.9.	DISEÑO DE LA HERRAMIENTA.	60
4.9.1.	Diseño del menú principal.	61
4.9.2.	Diseño de la barra de herramientas.....	62
4.9.3.	Diseño de la barra de navegación y menús de edición.....	66
4.9.4.	Diseño del cuadro que almacena la ruta.	70
5.	DESCRIPCIÓN DE LOS MODELOS 3D.	71
5.1.	Carrito de Golf Freedom SE.	72
5.2.	Suelo del campus de Leganés.	72
5.3.	Edificio Sabatini.....	73
5.4.	Edificio Betancourt.	74
5.5.	Edificio Padre Soler.....	76
5.6.	Edificio Rey Pastor.	77
5.7.	Edificio Torres Quevedo.	78
5.8.	Edificio Juan Benet.	80
5.9.	Polideportivo Alfredo Di Stefano.	81
5.10.	Caldera.	82
5.11.	Vegetación	83
5.12.	Bancos.	84
5.13.	Señales de tráfico.....	85
6.	EVALUACIÓN Y PRUEBAS.	88
6.1.	Pruebas de los modelos 3D.	88
6.2.	Pruebas del software.....	88
7.	CONCLUSIONES Y FUTUROS DESARROLLOS.	90
7.1.	Ampliaciones futuras:	90



7.2. Conclusiones.....	91
BIBLIOGRAFÍA.	93
ANEXOS.	94

1. INTRODUCCIÓN:

1.1. *ESCENARIO DEL PROYECTO.*

Los departamentos de Informática y de Industriales de la escuela Politécnica de la Universidad Carlos III de Madrid están desarrollando un proyecto de investigación centrado en técnicas de fusión sensorial, donde la plataforma de experimentación consiste en un carrito de golf equipado con una serie de sensores, cada uno de los cuales estará encargado de recopilar un determinado tipo de información. Esta a su vez enviara a una unidad central, la cual se encargará de procesar toda esta información y a partir de ella desarrollar sistemas de asistencia a la conducción. [La referencia del proyecto: Sistema de Detección de Peatones, Ciclistas y Motoristas - Fusión Sensorial. Plan Nacional de I+D+I. Ministerio de Educación y Ciencia. Investigador. Referencia TRA-2007-67374-C02-02. 2007-2009.]

Dentro de este proyecto se tiene ya desarrollado un software que mediante imágenes captadas por una cámara, es capaz de detectar y distinguir las distintas señales verticales de tráfico que se pueden encontrar en la vía pública. También se pretende desarrollar un software más complejo que sea capaz de identificar otros vehículos, ciclistas y viandantes. En este caso la aplicación es más compleja ya que hay una gran variedad en formas y tamaños de estos objetos, mientras que en el caso anterior estos parámetros son menos variables.

Antes de implementarlo y probarlo en el mundo real, se pretende realizar una serie de pruebas en un entorno simulado. El entorno de pruebas que vamos a utilizar es Microsoft Robotics Developer Studio, que es un entorno basado en Windows para el desarrollo de aplicaciones de robótica.

En esta parte del proyecto es en la que se integra este proyecto de fin de carrera, que consiste en modelar, un carrito de golf y otras entidades en tres dimensiones (3D) mediante un software con el que podamos importar el resultado e integrarlo en Robotics Developer Studio, para así poder realizar pruebas sobre el correcto funcionamiento del software ya desarrollado.

1.2. OBJETIVOS.

Este proyecto se centra en la generación y modificación de un entorno virtual 3D, que se asemeje al campus de Leganés, en el que se puedan realizar pruebas al software generado para la detección de peatones y señales de tráfico (Proyecto ASISTENTUR [Referencia: Sistema Avanzado de Asistencia a la Conducción para Entornos Urbanos. Financiado por el Ministerio de Educación y Ciencia a través del proyecto de la CICYT (2004-07), TRA2004-07441-C03-02/IA.]).

Para ello se generará una serie de modelos 3D, que serán los siguientes:

- Carrito de Golf Freedom SE.
- Suelo del campus de Leganes.
- Edificio Padre Soler.
- Edificio Betancourt.
- Edificio Rey Pastor.
- Edificio Sabatini.
- Edificio Torres Quevedo.
- Polideportivo Alfredo Di Stefano.
- Caldera.
- Personas variadas.
- Vegetación (Árboles, arbustos,...)
- Señales de tráfico (Horizontales y verticales).

Todos estos modelos se generarán a escala real.

Una vez generados, se introducirán todos ellos en un simulador de entornos virtuales para la obtención de un entorno, lo mas parecido posible al campus de Leganés, durante un día lectivo.

Finalizado todo esto, se creará una aplicación en la que se pueda modificar este entorno a gusto del usuario, con lo que las pruebas de software puedan ser lo más realistas posible. Estas posibles modificaciones se detallarán posteriormente.

Poder generar diferentes entornos en base a diversos parámetros es muy útil para la investigación en visión artificial y robótica, ya que gracias a ello podremos realizar pruebas particulares para cada software de detección, por ejemplo, generar un campus sólo con señales de tráfico para el proyecto de detección de señales (Proyecto ASISTENTUR [Referencia: Sistema Avanzado de Asistencia a la Conducción para Entornos Urbanos. Financiado por el Ministerio de Educación y Ciencia a través del proyecto de la CICYT (2004-07), TRA2004-07441-C03-02/IA.]), o uno solo con personas para el proyecto de detección de personas [REF.: Sistema de Detección de Pateones, Ciclistas y Motoristas - Fusión Sensorial. Plan Nacional de I+D+I. Ministerio de Educación y Ciencia. Investigador. Referencia TRA-2007-67374-C02-02. 2007-2009.]. Debido a la gran cantidad de parámetros que se pueden añadir o eliminar,

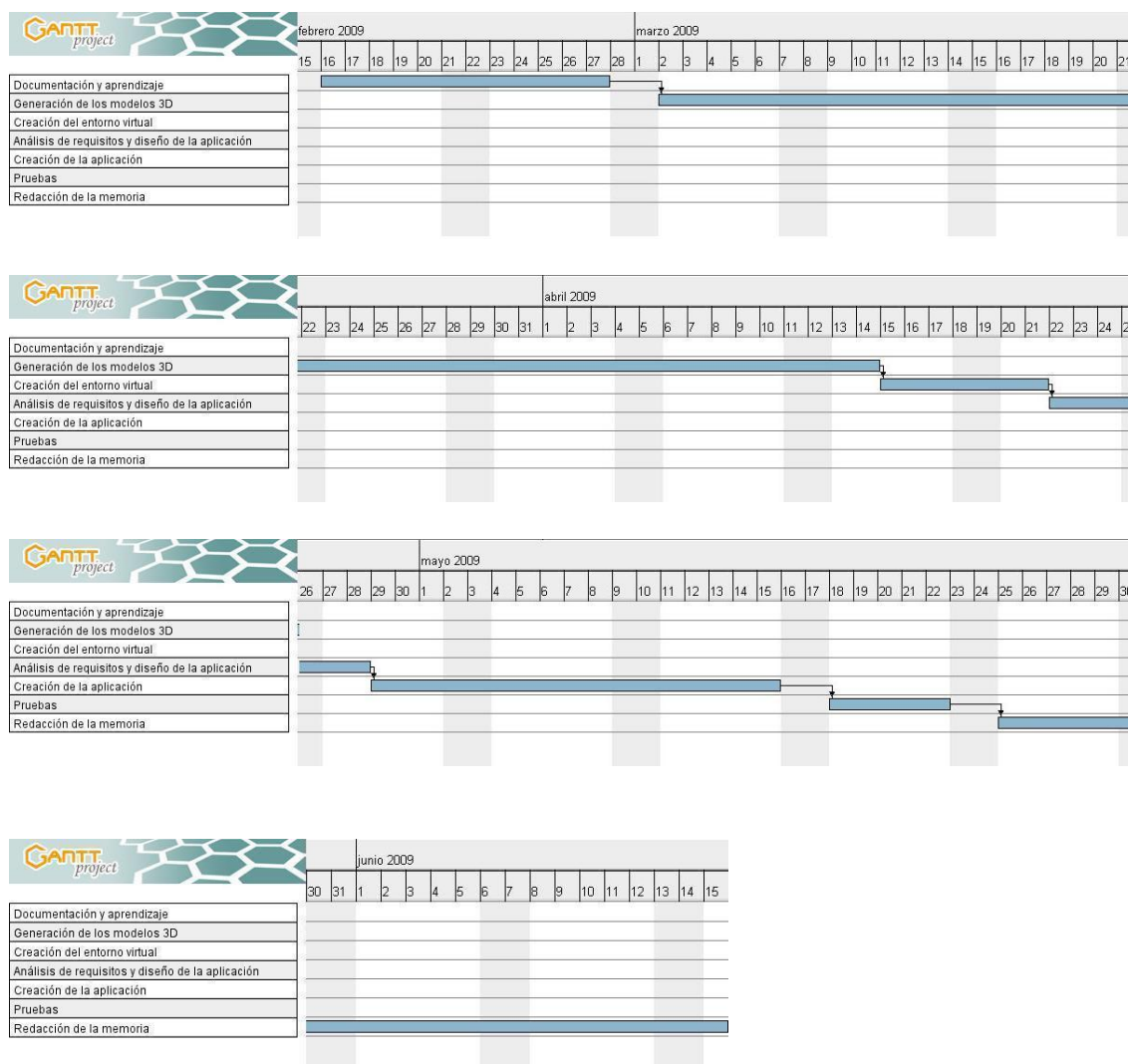


también se podrán realizar pruebas de las distintas versiones del software, es decir, si dentro del proyecto de detección de señales solo se ha desarrollado el apartado de detección de señales horizontales, se podrá generar un entorno particular con solo este tipo de señales.

También será muy útil poder comprobar como se comportan las cámaras y sensores con distintas cantidades de luz.

1.3. GESTIÓN DEL PROYECTO.

A continuación mostramos la planificación estimada para la ejecución de las tareas del proyecto mediante un diagrama de Gantt.



- **Documentación y aprendizaje:** En este apartado del proyecto nos centraremos en leer sobre la visión artificial y la robótica. Probaremos los distintos softwares de modelado 3D para ver cual de todos ellos se ajusta más a nuestras necesidades. Una vez realizado esto nos centraremos en aprender a manejar el software seleccionado.
- **Generación de los modelos 3D:** En esta etapa del proyecto nos centraremos en generar, mediante el software de modelado 3D seleccionado, los modelos necesarios para nuestro proyecto.



- **Creación del entorno virtual:** Aquí nos centraremos en exportar nuestros modelos a un formato compatible con Microsoft Robotics Developer Studio, así como la integración de todos los modelos dentro del simulador de entornos virtuales, para la generación del campus virtual.
- **Análisis de requisitos y diseño de la aplicación:** Se realizará un análisis de requisitos para ver las necesidades de nuestra aplicación. También se estudiará el diseño de la interfaz para que sea lo más intuitiva y usable posible.
- **Creación de la aplicación:** En esta etapa implementaremos la aplicación.
- **Pruebas:** Realizaremos distintas pruebas a la aplicación desarrollada, así como a los modelos 3D generados.
- **Redacción de la memoria:** Redactaremos la memoria.

1.4. ESTRUCTURA DE LA MEMORIA.

La memoria tiene tres partes diferenciadas que describiremos a continuación:

1.- Primera parte: En ella se detallarán las tecnologías y herramientas utilizadas para el desarrollo del proyecto, así como un análisis comparativo para demostrar los beneficios de haber elegido utilizar unas herramientas en vez de otras.

Esta parte la componen los siguientes capítulos:

- 1) Introducción.
- 2) ¿Por qué Sketchup?.
- 3) Herramientas y tecnologías utilizadas.

2.- Segunda parte: En esta parte se realizará una descripción de cada uno de los elementos 3D realizados. También se hará un análisis detallado de la aplicación que se va a desarrollar, los requisitos de usuario, casos de uso y otra serie de aspectos que se mostrarán mediante diagramas UML.

Esta parte la componen los siguientes capítulos:

- 4) Análisis de requisitos y diseño del software.
- 5) Descripción de los modelos 3D.

3.- Tercera parte: Se tratarán temas finales, como las pruebas, ampliaciones futuras para desarrollar y las conclusiones.

Esta parte la componen los siguientes capítulos:

- 6) Pruebas.
- 7) Conclusiones y futuros desarrollos.



2. ¿POR QUÉ SKETCHUP?

Para seleccionar la herramienta de modelado 3D en la que vamos a generar nuestros modelos, debemos centrarnos en estos requisitos:

- Poco periodo de aprendizaje: al ser, el generar los modelos, solo un apartado de nuestro proyecto, no podemos perder demasiado tiempo aprendiendo a utilizar la herramienta.
- Suficientemente potente para que los modelos que generemos se asemejen a la realidad, tanto en forma como en tamaño.
- Tenga la posibilidad de exportar los modelos al formato .obj, que es el que soporta Microsoft Robotics Studio Developer (MRSD).
- Que al introducir los modelos en MRSD, la física del modelo se ajuste el modelo.
- Por último, uno de los requisitos más importantes es el coste de la herramienta. Debe ser gratuita o que haya sido adquirida previamente por la universidad, ya que no disponemos de presupuesto.



2.1. INTRODUCCIÓN A LAS HERRAMIENTAS DE MODELADO 3D.

Muchos diseñadores necesitan llevar a cabo el desarrollo de sus productos, para los cual requieren de un buen software que les permita obtener todo el potencial de la herramienta.

Pero... ¿cuál escoger?, dependerá sobre todo del tipo de diseño que se desee hacer, para ello hay distintos tipos de software:











- Animación.
- Diseño Industrial.
- Ingeniería.
- Modelado artístico.
- Render de escenas.

A continuación realizaremos una breve orientación para mostrar los distintos tipos de herramientas que hay.

2.2. ANÁLISIS COMPARATIVO DE LAS DISTINTAS HERRAMIENTAS DE MODELADO 3D.

A continuación se presenta un análisis comparativo sobre los distintos tipos de software que hay para el modelado en 3D, para ver cual puede ser el que mejor se ajusta a nuestros requisitos.



	Autodesk 3ds max 9 SP2	Autodesk Maya complete 8.5	Blender 3D 2.45	SolidWorks	Sketchup Pro 6	Sketchup Pro 6 (Free Version)	Autodesk Softimage Mod Tool	Autodesk XSI	AC3D	Solid Edge
Mercado	Juegos	Juegos - Películas	Juegos - Películas	Modelado mecánico	Educación	Educación	Juegos	Juegos y efectos visuales	Juegos	Modelado mecánico
Precio Final	5,000 €	2,500 €	Gratis	7,500 €	331 €	Gratis	Gratis	2600 €	80 €	3578 €
Suscripción Anual	Si	Si	No	Si	No	No	Si	Si	No	Si
Plataformas										
Versión de prueba	Si	Si	No	Si	Si	Si	Si	Si	Si	Si
Idiomas	FR, EN	EN, JPN	EN	EN	EN ESP	EN ESP	EN	EN, JPN	EN	EN ESP
Idiomas incorporados	No	No	Todos incluidos	No	No	No	No	No	No	No
Popularidad en la industria	Muy buena	Muy buena	Baja	Buena	Normal	Normal	Norma	Muy buena	Normal	Muy buena
Tiempo de aprendizaje	< 2 meses	< 3 meses	< 3 meses	< 3 meses	< 1 mes	< 1 mes	< 2 meses	< 3 meses	< 2 mes	< 3 meses
Soporte para usuarios simples	Regular	Bueno	No tiene	Regular	Bueno	Bueno	No tiene	Bueno	Bueno	Bueno
Interfaz	Limpia y potente	Flexible, potente, pero no intuitiva	Poco intuitiva	Práctica e intuitiva	Flexible y muy intuitiva	Flexible y muy intuitiva	Práctica e intuitiva	Flexible, potente, pero no intuitiva	Flexible y muy intuitiva	Potente y flexible.
Documentación	Buena	Excelente	Buena	Buena	Regular	Regular	Buena	Excelente	Buena	Excelente
DVD de entrenamiento	Buena	Muy buena	Regular	Buena	Excelente	Excelente	No tiene	No tiene	No tiene	Buena



2.2.1. Autodesk 3ds Max 9 SP2:

Autodesk 3D Studio Max es un programa de creación de gráficos y animación 3D desarrollado por Autodesk Media & Entertainment (Anteriormente conocidos como Discreet y Kinetix). Fue desarrollado como sucesor para sistemas operativos Win32 del 3D Studio creado para DOS. Kinetix fue más tarde fusionada con la última adquisición de Autodesk, Discreet Logic.

3d Studio Max es uno de los programas de animación 3D más utilizados. Dispone de una sólida capacidad de edición, una omnipresente arquitectura de plugins y una larga tradición en plataformas Microsoft Windows. 3ds Max es utilizado en mayor medida por los desarrolladores de videojuegos, aunque también en el desarrollo de proyectos de animación como películas o anuncios de televisión, efectos especiales y en arquitectura.

Desde la primera versión 1.0 hasta la 4.0 el programa pertenecía a Autodesk con el nombre de 3d Studio. Más tarde, Kinetix compró los derechos del programa y lanzó 3 versiones desde la 1.0 hasta la 2.5 bajo el nombre de 3d Studio Max. Más tarde, la empresa Discreet compró los derechos, retomando la familia empezada por Autodesk desde la 4.0 hasta 6.0 también bajo el nombre de 3d Studio Max. Finalmente, Autodesk retomó el programa desarrollándolo desde la versión 7.0 en adelante bajo el mismo nombre, hasta la versión 9. A partir de ésta, se denomina Autodesk 3d Studio Max.

Este programa es uno de los más reconocidos modeladores de 3d masivo, habitualmente orientado al desarrollo de videojuegos, con el que se han hecho enteramente títulos como las sagas 'Tomb Raider', 'Splinter Cell' y una larga lista de títulos de la empresa Ubisoft.



2.2.2. *Autodesk Maya:*

Maya es un programa informático dedicado al desarrollo de gráficos en 3d, efectos especiales y animación. Surgió a partir de la evolución de Power Animator y de la fusión de Alias y Wavefront, dos empresas canadienses dedicadas a los gráficos generados por ordenador. Más tarde Silicon Graphics (ahora SGI), el gigante informático, absorbió a Alias-Wavefront, que finalmente ha sido absorbida por Autodesk.

Maya se caracteriza por su *potencia y las posibilidades de expansión y personalización de su interfaz y herramientas*. MEL (Maya Embedded Language) es el código que forma el núcleo de Maya, y gracias al cual se pueden crear scripts y personalizar el paquete.

Maya posee numerosas herramientas para modelado, animación, render, simulación de ropa y cabello, dinámicas (simulación de fluidos), etc.

Además Maya es el único software de 3D acreditado con un Oscar gracias al enorme impacto que ha tenido en la industria cinematográfica como herramienta de efectos visuales, con un uso muy extendido debido a su gran capacidad de ampliación y personalización.

2.2.3. *Blender:*

Blender es un programa multiplataforma, dedicado especialmente al modelado, animación y creación de gráficos tridimensionales. El programa fue inicialmente distribuido de forma gratuita pero sin el código fuente, con un manual disponible para la venta, aunque posteriormente pasó a ser software libre. Actualmente es compatible con todas las versiones de Windows, Mac OS X, Linux, Solaris, FreeBSD e IRIX.

Tiene una muy peculiar interfaz gráfica de usuario, que se critica como poco intuitiva, pues no se basa en el sistema clásico de ventanas; pero tiene a su vez ventajas importantes sobre éstas, como la configuración personalizada de la distribución de los menús y vistas de cámara.



2.2.4. *SolidWorks:*

SolidWorks es un programa de diseño asistido por computador para modelado mecánico que corre bajo el sistema operativo Microsoft Windows y es desarrollado en la actualidad por SolidWorks Corp., una subsidiaria de Dassault Systèmes (Suresnes, Francia). Es un modelador de sólidos paramétrico, que usa el kernel de modelado geométrico Parasolid. Fue introducido en el mercado en 1995 para competir con otros programas CAD como Pro/ENGINEER, I-DEAS, Unigraphics, CATIA, y Autodesk Mechanical Desktop y es actualmente el líder del mercado del modelado mecánico en CAD.

El programa permite modelar piezas y conjuntos y extraer de ellos tanto planos como otro tipo de información necesaria para la producción. Es un programa que funciona con base en las nuevas técnicas de modelado con sistemas CAD. El proceso consiste en trasvasar la idea mental del diseñador al sistema CAD, "construyendo virtualmente" la pieza o conjunto. Posteriormente todas las extracciones (planos y ficheros de intercambio) se realizan de manera bastante automatizada.

La empresa SolidWorks Corp. fue fundada en 1993 por Jon Hirschtick con su sede en Concord, Massachusetts y lanzó su primer producto, SolidWorks 95, en 1995. En 1997 Dassault Systèmes, mejor conocida por su software CAD CATIA, adquirió la compañía. Actualmente posee el 100% de sus acciones y es liderada por Jeff Ray.

SolidWorks ha entrado con fuerza al mercado latinoamericano, especialmente en América del Sur. Algunos de sus distribuidores autorizados latinos se encuentran ubicados en México, República Dominicana, Colombia, Venezuela, Perú, Chile, Honduras y Argentina.

2.2.5. *Sketchup:*

SketchUp es un programa computacional de modelaje y diseño en 3D (tercera dimensión) encaminado a la arquitectura, desarrollo de videojuegos, películas, ingeniería civil o simple entretenimiento personal.

También se ha comenzado a trabajar con pre-adolescentes para despertar en ellos el interés de crear y usar esta herramienta. Los edificios creados con el programa pueden ser geo-referenciados y colocados sobre las imágenes de Google Earth. Los modelos pueden ser subidos a la red mediante el propio programa Google SketchUp y directamente almacenarse en la base de datos 3d Warehouse para ser compartidos.

SketchUp fue diseñado para usarlo de una manera intuitiva y flexible, facilitando ampliamente su uso en comparación con otros programas de modelado 3D. Cualquier persona, desde un niño hasta un adulto, pueden de manera muy sencilla aprender a utilizar esta herramienta para diseño tridimensional. El programa también incluye en sus recursos un tutorial en vídeo para ir aprendiendo paso a paso cómo se puede ir diseñando y modelando el propio ambiente. Se recomienda a los principiantes tener paciencia para seguir las explicaciones del tutorial (está disponible la versión en español). Como una introducción, el tutorial tiene una importancia básica para comprender lo sencillo, práctico y poderoso que en definitiva resulta ser este programa.

SketchUp permite moldear imágenes en 3D de edificios, coches, personas y cualquier objeto o artículo dentro de la imaginación del diseñador o dibujante. Además, para facilidad, el programa incluye una galería de objetos, texturas e imágenes para descargar.

2.2.6. *Softimage Mod Tool:*

Es una herramienta gratuita para realizar modelos 3D para los juegos, ésta es una alternativa a tener en cuenta, además de tener varias opciones para integrarla con XNA.

Acaban de anunciar una nueva actualización que llegará en Agosto con una lista de novedades bastante atractiva, para ver más información y para la descarga del programa (totalmente gratuito).



2.2.7. *Softimage XSI:*

El Autodesk ® Softimage ® 7.5 (comúnmente SOFTIMAGE|XSI) es un software completo de 3D para realizar efectos visuales y producción de juegos. Construido sobre una arquitectura nueva, sumamente extensible.

Softimage features Interactive Creative Environment (ICE), una nueva plataforma para construir efectos e instrumentos visualmente que usan un paradigma a base de nodo. El impulso de ICE, que es un motor de procesamiento en paralelo de alto rendimiento que aprovecha terminales de trabajo multiprincipales, dándole el funcionamiento extremo y la adaptabilidad. El resultado es un salto gigantesco adelante en la eficacia de producción y la consistencia que culmina en normas de producción más altas.

Softimage features ICE (Interactive Creative Environment), es una tecnología innovadora que transforma el software en una plataforma poderosa y flexible abierta.

2.2.8. *AC3D:*

AC3D es una completa herramienta de diseño 3D multiplataforma.

Con un interfaz sencillo pero repleto de funciones, podremos crear impresionantes modelos en tres dimensiones y personalizarlos fácilmente con el sistema de arrastrar y soltar.

AC3D cuenta además con varias vistas para no perder detalle en ningún momento, permite navegar cómodamente por nuestras creaciones y cuenta con soporte para los formatos habituales.

2.2.9. *Solid Edge:*

Solid Edge es un programa de parametrizado de piezas en 3D basado en un software de sistema de diseño asistido por ordenador (CAD). Permite el modelado de piezas de distintos materiales, doblado de chapas, ensamblaje de conjuntos, soldadura y funciones de dibujo en plano para ingenieros.

Este es uno de los paquetes instalados a enterrar el uso masivo del CAD 2D dando paso al CAD 3D, con las consiguientes ventajas a todos los niveles del trabajo. A través de software de terceras partes, es compatible con otras



tecnologías PLM. También trae "Insight", escrito en PDM y con funcionalidades CPD basadas en tecnología Microsoft.

2.3. CONCLUSIONES SOBRE EL ANÁLISIS Y EL PORQUÉ DE ELEGIR SKETCHUP.

- Una de las conclusiones a las que hemos llegado es que sobre todo lo leído e investigado, sin duda el software de pago que más se usa en el mundo es 3ds Max, Maya y XSI.

3DS Max se usa sobre todo en el mundo de los videojuegos y la arquitectura, mientras que en el mundo del cine quien manda es Maya.

XSI sin embargo ha ido ganando terreno a los 2 anteriores, pero sigue teniendo menos mercado, aunque eso no quiere decir que sea peor, todo lo contrario, es el software con el core más nuevo.

Como conclusión decir que estos 3 softwares son muy parecidos y no ha nada que se pueda hacer en uno que no puedas hacer en el otro. A la hora de elegir el software la empresa tiene que hacer un balance de pros y contras para decantarse por uno o por otro.

- En el apartado de software gratis, Blender se hace cada vez más grande y ya empieza a competir con los de pago, lo mismo que Sketchup (Versión gratuita).

Por todo lo mencionado anteriormente se ha elegido la versión gratuita de Sketchup para realizar los modelos 3D del proyecto, ya que las limitaciones de la versión gratuita no suponen ningún problema para generar los modelos correctamente. Otro punto a favor es el poco tiempo de aprendizaje que se necesita, ya que su interfaz es tremendamente intuitiva, y el software es suficientemente potente para poder generar nuestros modelos a la perfección. También decir que permite exportar los modelos al formato .OBJ, que es el formato que soporta Microsoft Robotics Studio, para generar el entorno virtual. Este software se ajusta perfectamente a nuestras necesidades.

3. HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS.

3.1. *Visual Studio 2008:*

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión net 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

Visual Studio 2008 fue publicado (RTM) el 17 de Noviembre de 2007 en inglés, mientras que la versión en castellano no fue publicada hasta el 2 de Febrero de 2008.

El nuevo .NET framework (versión 3.5) está diseñado para aprovechar las ventajas que ofrece el nuevo sistema operativo "Windows Vista" a través de sus subsistemas "Windows Communication Foundation" (WCF) y "Windows Presentation Foundation" (WPF). El primero tiene como objetivo la construcción de aplicaciones orientadas a servicios mientras que el último apunta a la creación de interfaces de usuario más dinámicas que las conocidas hasta el momento.

A las mejoras de desempeño, escalabilidad y seguridad con respecto a la versión anterior, se agregan entre otras, las siguientes novedades.

- La mejora en las capacidades de Pruebas Unitarias permiten ejecutarlas más rápido independientemente de si lo hacen en el entorno IDE o desde la línea de comandos. Se incluye además un nuevo soporte para diagnosticar y optimizar el sistema a través de las herramientas de pruebas de Visual Studio. Con ellas se podrán ejecutar perfiles durante las pruebas para que ejecuten cargas, prueben procedimientos contra un sistema y registren su comportamiento; y utilizar herramientas integradas para depurar y optimizar.
- Con Visual Studio Tools for Office (VSTO) integrado con Visual Studio 2008 es posible desarrollar rápidamente aplicaciones de alta calidad basadas en la interfaz de usuario (UI) de Office que personalicen la experiencia del usuario y mejoren su productividad en el uso de Word, Excel, PowerPoint, Outlook, Visio, InfoPath y Project. Una completa compatibilidad para implementación con ClickOnce garantiza el



entorno ideal para una fácil instalación y mantenimiento de las soluciones Office.

- Visual Studio 2008 permite incorporar características del nuevo Windows Presentation Foundation sin dificultad tanto en los formularios de Windows existentes como en los nuevos. Ahora es posible actualizar el estilo visual de las aplicaciones al de Windows Vista debido a las mejoras en Microsoft Foundation Class Library (MFC) y Visual C++. Visual Studio 2008 permite mejorar la interoperabilidad entre código nativo y código manejado por .NET. Esta integración más profunda simplificará el trabajo de diseño y codificación.
- LINQ (Language Integrated Query) es un nuevo conjunto de herramientas diseñado para reducir la complejidad del acceso a Base de Datos, a través de extensiones para C++ y Visual Basic así como para Microsoft .NET Framework. Permite filtrar, enumerar, y crear proyecciones de muchos tipos y colecciones de datos utilizando todos la misma sintaxis, prescindiendo del uso de lenguajes especializados como SQL o XPath.
- Visual Studio 2008 ahora permite la creación de soluciones multiplataforma adaptadas para funcionar con las diferentes versiones de .Net Framework: 2.0. (Incluido con Visual Studio 2005), 3.0 (incluido en Windows Vista) y 3.5 (incluido con Visual Studio 2008).
- .NET 3.5 incluye biblioteca ASP.NET AJAX para desarrollar aplicaciones web más eficientes, interactivas y altamente personalizadas que funcionen para todos los navegadores más populares y utilicen las últimas tecnologías y herramientas Web, incluyendo Silverlight y Popfly.

3.2. *Robotics Developer Studio:*

Microsoft Robotics Studio es un entorno basado en Windows para el control de robótico y la simulación.

Características:

Plataforma de desarrollo robótica extremo-a-extremo
Herramienta de programación visual para crear y depurar aplicaciones robóticas. El desarrollador puede interactuar con los robots mediante interfaces basadas en web o en Windows.

Simulación 3-D
La simulación realística está provista por el motor PhysX de AGEIA. Se posibilita la emulación por software o la aceleración por hardware.

Entorno de ejecución ligero orientado a servicios
El desarrollador puede acceder fácilmente a los sensores y actuadores de los robots, proporcionada por una librería de implementación de concurrencia basada en .NET. La comunicación está basada en mensajes, permitiendo la comunicación entre módulos.

Plataforma escalable y extensible
La propia aplicación es multi-plataforma-robótica. Se permiten varios lenguajes - lenguajes Microsoft Visual Studio Express languages (Visual C#® y Visual Basic® .NET), JScript® y IronPython 1.0 Beta 1, y lenguajes de terceras partes que se adecuen a la arquitectura basada en servicios.



3.3. *Netbeans:*

NetBeans se refiere a una plataforma para el desarrollo de aplicaciones de escritorio usando Java y a un entorno de desarrollo integrado (IDE) desarrollado usando la Plataforma NetBeans.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados *módulos*. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio 2000 y continúa siendo

Un proyecto de código abierto no es nada más ni nada menos que un proceso. Toma tiempo encontrar el equilibrio. El primer año, fue crucial como inicio. Los dos años siguientes, se orientó hacia código abierto. Como muestra de lo abierto que era, en los primeros dos años había más debate que implementación.

Con NetBeans 3.5 se mejoró enormemente en desempeño, y con la llegada de NetBeans 3.6, se reimplementó el sistema de ventanas y la hoja de propiedades, y se limpió enormemente la interfaz. NetBeans 4.0 fue un gran cambio en cuanto a la forma de funcionar del IDE, con nuevos sistemas de proyectos, con el cambio no solo de la experiencia de usuario, sino del reemplazo de muchas piezas de la infraestructura que había tenido NetBeans anteriormente. NetBeans IDE 5.0 introdujo un soporte mucho mejor para el desarrollo de nuevos módulos, el nuevo constructor intuitivo de interfaces Matisse, un nuevo y rediseñado soporte de CVS, soporte a Sun ApplicationServer 8.2, Weblogic9 y JBoss 4.

Con Netbeans 6.01 y 6.5 Se dio soporte a frameworks comerciales como son Struts, Hibernate.



3.4. Java.

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un *bytecode*, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el *bytecode* es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del *bytecode* por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrollados por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Entre noviembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre (aunque la biblioteca de clases de Sun que se requiere para ejecutar los programas Java todavía no es software libre).



3.5. C#.

C# (pronunciado "*si sharp*" en inglés y "*c sostenido*" o "*c almohadilla*" en español) es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA e ISO.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma.NET el cual es similar al de Java aunque incluye mejoras derivadas de otros lenguajes (entre ellos Delphi).

En el caso del lenguaje C#, la intención original de los creadores del lenguaje fue llamarlo "C en re bemol", para denotar belleza e inteligencia en el diseño de tal lenguaje. Debido a que se encontró parecido el símbolo "sharp" o "numeral" en español, al correspondiente de Do sostenido mayor, el nombre del lenguaje quedó como C#.

C#, como parte de la plataforma.NET, está normalizado por ECMA desde diciembre de 2001 (ECMA-334 "Especificación del Lenguaje C#"). El 7 de noviembre de 2005 salió la versión 2.0 del lenguaje que incluía mejoras tales como tipos genéricos, métodos anónimos, iteradores, tipos parciales y tipos anulables. El 19 de noviembre de 2007 salió la versión 3.0 de C# destacando entre las mejoras los tipos implícitos, tipos anónimos y el LINQ (Language Integrated Query).

Aunque C# forma parte de la plataforma.NET, ésta es una interfaz de programación de aplicaciones (**API**); mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el Framework de DotGNU - Mono que genera programas para distintas plataformas como Win32, UNIX y Linux.

3.6. *Google Earth.*

Google Earth es un programa informático similar a un Sistema de Información Geográfica (SIG), creado por la empresa Keyhole Inc., que permite visualizar imágenes en 3D del planeta, combinando imágenes de satélite, mapas y el motor de búsqueda de Google que permite ver imágenes a escala de un lugar específico del planeta.

Google Earth permite introducir el nombre de un hotel, colegio o calle y obtener la dirección exacta, un plano o vista del lugar. También se puede visualizar imágenes vía satélite del planeta. También ofrece características 3D como dar volumen a valles y montañas, y en algunas ciudades incluso se han modelado los edificios. La forma de moverse en la pantalla es fácil e intuitiva, con cuadros de mandos sencillos y manejables.

Además, es posible compartir con otros usuarios enlaces, medir distancias geográficas, ver la altura de las montañas, ver fallas o volcanes y cambiar la vista tanto en horizontal como en vertical.

Google Earth también dispone de conexión con GPS (Sistema de Posicionamiento Global) , alimentación de datos desde fichero y base de datos en sus versiones de pago.

También tiene un Simulador de Vuelo de google earth bastante real con el que podremos sobrevolar cualquier lugar del planeta.

La versión 4 ha incorporado notables mejoras:

- Interfaz en inglés, español, francés y alemán.
- Tener relación con Google SketchUp, un programa de modelaje 3D desde el cual se pueden subir modelos 3D de edificios a Google Earth.
- Panel de mandos que interfiere más discreto y gana en espacio para la visualización de imágenes.
- Mejoras que permiten ver imágenes en 3D "texturizadas" (superficies más realistas, ventanas, ladrillos...)
- Versión en los tres sistemas operativos más importantes para computadores personales (Windows, Linux, y MAC)

Inclusión de enlaces a los artículos de la Wikipedia en inglés en ciudades, monumentos, accidentes geográficos y otros puntos de interés.



3.7. *Sketchup.*

SketchUp es un programa computacional de modelaje y diseño en 3D (tercera dimensión) encaminado a la arquitectura, desarrollo de videojuegos, películas, ingeniería civil o simple entretenimiento personal.

También se ha comenzado a trabajar con pre-adolescentes para despertar en ellos el interés de crear y usar esta herramienta. Los edificios creados con el programa pueden ser geo-referenciados y colocados sobre las imágenes de Google Earth. Los modelos pueden ser subidos a la red mediante el propio programa Google SketchUp y directamente almacenarse en la base de datos 3d Warehouse para ser compartidos.

SketchUp fue diseñado para usarlo de una manera intuitiva y flexible, facilitando ampliamente su uso en comparación con otros programas de modelado 3D. Cualquier persona, desde un niño hasta un adulto, pueden de manera muy sencilla aprender a utilizar esta herramienta para diseño tridimensional. El programa también incluye en sus recursos un tutorial en vídeo para ir aprendiendo paso a paso cómo se puede ir diseñando y modelando el propio ambiente. Se recomienda a los principiantes tener paciencia para seguir las explicaciones del tutorial (está disponible la versión en español). Como una introducción, el tutorial tiene una importancia básica para comprender lo sencillo, práctico y poderoso que en definitiva resulta ser este programa.

SketchUp permite moldear imágenes en 3D de edificios, coches, personas y cualquier objeto o artículo dentro de la imaginación del diseñador o dibujante. Además, para facilidad, el programa incluye una galería de objetos, texturas e imágenes para descargar.



4. ANALISIS DE REQUISITOS Y DISEÑO DEL SOFTWARE

4.1. INTRODUCCION.

Para el diseño de esta aplicación deberemos basarnos en un estándar de Ingeniería de Software que tenga unos sólidos cimientos y una fuerte estabilidad. Para este proyecto haremos uso del estándar ESA PSS-05-0, utilizado por la Agencia Espacial Europea, ya que cumple con nuestras expectativas. Este estándar se ha adaptado a un proyecto pequeño.

Dicho estándar puede consultarse en la página web oficial de la Agencia Espacial Europea, en las siguiente URL de Internet: <http://www.esa.int/esaCP/index.html>

Aplicando el estándar ESA PSS-05-0 seguiremos los siguientes pasos:

- 1) Requisitos de Usuario de Capacidad: Estos requisitos representan las capacidades que los usuarios necesitan que tenga el sistema para que pueda resolver sus problemas o cumplir sus objetivos.
- 2) Requisitos de Usuario de Restricción: Estos requisitos representan las restricciones que los usuarios establecen acerca de cómo se solucionarán los problemas o se lograrán los objetivos.
- 3) Diagramas de Casos de Uso: Estos proporcionan uno o más escenarios que indican cómo deberían interactuar el sistema con los usuarios o con otros sistemas para conseguir un objetivo específico.
- 4) Diagrama de Secuencia: Muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase.
- 5) Trazabilidad: Capturar las relaciones de implementación y dependencia en el modelo, siguiendo un histórico de los pasos seguidos para desarrollar el software.

4.1.1. Alcance del software

El propósito fundamental de este software es poder modificar un entorno 3D ya diseñado, basado en el campus de Leganes de la Universidad Carlos III de Madrid, sin la necesidad de tener conocimientos sobre programación de dichos entornos. Estos escenarios se generarán a partir de las características seleccionadas que hará el usuario en la interfaz.

Las características posibles a modificar en el entorno, mediante el software son:

- Mobiliario.
- Personas.
- Señales.
- Edificios.
- Climatología.

Las modificaciones serán posibles en un principio sobre un número limitado de características, pudiendo en un futuro ampliarlo añadiendo nuevas posibles modificaciones.

Será sencillo de usar, con un funcionamiento fácil de aprender y recordar.

Como principal beneficio se podría destacar que el usuario generará de forma eficaz y sencilla, sin necesidad de programación, un entorno 3D que se ajuste a sus necesidades.

4.1.2. Visión General

El documento está compuesto por tres partes:

1. Introducción: Pretende dar una idea general sobre la importancia de este documento y de su lectura detallada por todas las partes e introducir los detalles más superficiales de la aplicación. Así mismo es en esta parte del documento donde se pretende facilitar la comprensión del resto del mismo.
2. Descripción general: Pretende realizar una descripción de los propósitos generales con los que la aplicación pretende cumplir. Cubre también aspectos relativos al usuario de la aplicación y al entorno de utilización de la misma. Describe también las posibles dependencias de otras aplicaciones o sistemas.
3. Especificación de requisitos: Enumeración detallada de objetivos descritos con la máxima concreción posible. Estos requisitos están ordenados de forma jerárquica, se distinguen las categorías de

requisitos de capacidad y de restricción. Es importante comprender que en esta parte del documento se pretenden describir todas las características del software por lo cual **todo aquello que no esté reflejado aquí no será implementado.**

4.2. DESCRIPCIÓN GENERAL.

4.2.1. *Perspectiva del producto*

Actualmente los departamentos de Informática y de Industriales de la escuela Politécnica de la Universidad Carlos III de Madrid están desarrollando un proyecto de investigación centrado en técnicas de fusión sensorial, donde la plataforma de experimentación consiste en un carrito de golf equipado con una serie de sensores, para comprobar que el software desarrollado para estos sensores funciona correctamente se necesita realizar una serie de pruebas dentro de un entorno simulado.

El objetivo es desarrollar un software muy específico y de manejo sencillo que permita modificar este entorno para la realización de las mismas.

Algunos de los proyectos relacionados son los siguientes:

- Proyecto ASISTENTUR [Referencia: Sistema Avanzado de Asistencia a la Conducción para Entornos Urbanos. Financiado por el Ministerio de Educación y Ciencia a través del proyecto de la CICYT (2004-07), TRA2004-07441-C03-02/IA.].
- Sistema de Detección de Peatones, Ciclistas y Motoristas - Fusión Sensorial. Plan Nacional de I+D+I. Ministerio de Educación y Ciencia. Investigador. Referencia TRA-2007-67374-C02-02. 2007-2009.

4.2.2. *Capacidades generales*

El producto se considera necesario pues no existe programa alguno que sea tan específico y sencillo como el que queremos desarrollar.

- Rápido aprendizaje
- Funcionamiento fácil de recordar
- Modificación del entorno 3D de forma sencilla e intuitiva

- Navegación rápida, eficaz y sencilla por el sistema.
- Consumo de pocos recursos.
- Multiplataforma.
- Facilidad de uso.
- Eficiencia.
- Portabilidad.
- Integridad.
- Fiabilidad.
- Centralizado.

Inicialmente el producto está dirigido a la modificación en el entorno 3D de:

- Mobiliario.
- Personas.
- Señales.
- Edificios.
- Climatología.

4.2.3. Restricciones generales

El sistema debe ofrecer un interfaz fácil de manejar, intuitivo y con un funcionamiento fácil de recordar.

4.2.4. Características de los usuarios

El usuario necesita conocimientos básicos de informática (aunque se presupone debido al público al que va dirigido este producto).

El idioma empleado será el español.

El usuario será una persona que quiera realizar modificaciones a un entorno 3D, para la realización de pruebas de software centrado en técnicas de fusión sensorial.

4.2.5. Entorno operacional

El sistema va a funcionar en PCs, con independencia del SO que se utilice. Estos PCs no requerirán que dispongan de una gran cantidad de recursos.

4.2.6. Suposiciones y dependencias

El sistema generará el entorno 3D deseado a partir de otro entorno completo ya existente y almacenado en un archivo XML.

El sistema guardará el entorno generado en la misma carpeta donde está almacenado el entorno principal.

El nombre del entorno principal deberá ser campus.xml

4.3. DEFINICIÓN DE REQUISITOS.

En este capítulo se exponen los requisitos de usuario y requisitos software de la aplicación que se va a desarrollar y que se utilizará para la simulación.

La definición de requisitos es fundamental para acotar el alcance de la aplicación y asegurar que se conseguirán los resultados deseados.

La definición de los requisitos de un sistema informático se realiza en dos pasos:

- 1) El primero consiste en entrevistar al cliente, en nuestro caso será con el cliente, el Departamento de Informática de la Universidad Carlos III de Madrid, y en concreto al grupo de personas integrante del proyecto de investigación centrado en técnicas de fusión sensorial, para concretar las ideas que éste pueda tener acerca de “**QUÉ**” tiene que hacer el sistema.
- 2) El segundo paso toma estas ideas y, gracias a la experiencia del analista informático, las transforma en la definición de “**COMO**” se construirá el sistema.

En el caso concreto de este proyecto, seguiremos el estándar de Ingeniería del Software: **ESA PSS-05-0**, utilizado por la Agencia Espacial Europea.

Dicho estándar puede consultarse en la página web oficial de la Agencia Espacial Europea, en la siguiente URL de Internet:
<http://www.esa.int/esaCP/index.html>

4.4. IDENTIFICACIÓN DE REQUISITOS.

La tarea de identificación de requisitos se lleva a cabo mediante entrevistas periódicas con el cliente.

Dichas entrevistas son entrevistas abiertas (diálogo fluido y espontáneo) o semi-abiertas, entre los ingenieros del software y el cliente, tomando nota de las necesidades que éste plantee.

Para que la recogida de requisitos se realice de forma clara, sencilla y estructurada se ha definido una plantilla con las siguientes propiedades:

Identificador Nombre	
Descripción	
Necesidad	Alta Media Baja
Prioridad	Absoluta Intermedia Básica
Fuente	Usuario
Estabilidad	Alta Baja

A continuación explicamos brevemente el significado de los parámetros que se recogen en la plantilla:

- ❖ **Identificador y Título:** Identifica de forma unívoca cada uno de los requisitos. Dicho identificador sigue la siguiente nomenclatura:

RU+Tipo+Número.

Donde:

- RU: Son las siglas de “Requisito de Usuario”.
 - Tipo: Puede tomar los valores “C”, si se trata de un requisito de usuario de Capacidad, y “R” si se trata de un requisito de usuario de Restricción.
 - Número: Será un número de tres cifras que empezará por 001 y se irá incrementado en una unidad por cada nuevo requisito añadido.
-
- ❖ **Nombre:** Expresa el nombre del requisito, en pocas palabras un resumen del requisito.
 - ❖ **Descripción:** Breve comentario textual del requisito.



- ❖ **Necesidad:** Indica el nivel de necesidad del requisito dentro del sistema final. Puede tomar los valores “Alta”, “Media” o “Baja”.
 - “Alta”: Cuando el cliente no acepte ninguna negociación.
 - “Media”: Cuando el requisito se pueda negociar.
 - “Baja”: Cuando su implementación puede ser eliminada.

- ❖ **Prioridad:** Indica la prioridad en el desarrollo del requisito. Puede tomar los valores “Absoluta”, “Intermedia” o “Basica”.

- ❖ **Fuente:** Indica el origen a partir del cual se ha obtenido el documento.

- ❖ **Estabilidad:** Indica la posibilidad de que el requisito cambien a lo largo del desarrollo de la aplicación. Puede tomar los siguientes valores:
 - Alta: Cuando el cliente asegura que no va a ser modificado.
 - Baja: Cuando el requisito puede variar en función de las sucesivas etapas del proyecto.

4.5. REQUISITOS DE USUARIO.

4.5.1. Requisitos de capacidad

4.5.1.1. URC01 Cargar Fichero Principal

URC01	Cargar Fichero Principal
Descripción	El programa debe ser capaz de cargar el fichero campus.xml, el cual almacena el entorno completo.
Necesidad	Alta.
Prioridad	Básica
Fuente	Usuario
Estabilidad	Alta

4.5.1.2. URC02 Editar Personas

URC02	Editar Personas
Descripción	El sistema será capaz de modificar las personas existentes en el entorno principal, a gusto del usuario. Éste podrá añadir o eliminar grupos de personas seleccionando opciones en el menú Personas.
Necesidad	Alta.
Prioridad	Básica
Fuente	Usuario
Estabilidad	Alta

4.5.1.3. URC03 Editar Señales

URC03	Editar Señales
Descripción	El sistema será capaz de modificar las señales existentes en el entorno principal, a gusto del usuario. Éste podrá añadir o eliminar grupos de señales, horizontales o verticales, seleccionando opciones en el menú Señales.
Necesidad	Alta.
Prioridad	Básica
Fuente	Usuario
Estabilidad	Alta

4.5.1.4. URC04 Editar Edificios

URC04	Editar Edificios
Descripción	El sistema será capaz de modificar los edificios existentes en el entorno principal, a gusto del usuario. Éste podrá añadir o eliminar edificios seleccionando opciones en el menú Edificios.
Necesidad	Alta.
Prioridad	Básica
Fuente	Usuario
Estabilidad	Alta

4.5.1.5. URC05 Editar Climatología

URC05	Editar Climatología
Descripción	El sistema será capaz de modificar la climatología del entorno principal, así como la cantidad de luz, a gusto del usuario. Éste podrá modificar la climatología seleccionando opciones en el menú Climatología.
Necesidad	Alta.
Prioridad	Básica
Fuente	Usuario
Estabilidad	Alta

4.5.1.6. URC06 Editar Mobiliario

URC04	Editar Mobiliario
Descripción	El sistema será capaz de modificar los edificios existentes en el entorno principal, a gusto del usuario. . Éste podrá añadir o eliminar edificios seleccionando opciones en el menú Edificios.
Necesidad	Alta.
Prioridad	Básica
Fuente	Usuario
Estabilidad	Alta

4.5.1.7. URC07 Almacenar Entorno Seleccionado

URC07	Almacenar Entorno Seleccionado
Descripción	El sistema será capaz de almacenar en un archivo .xml el entorno seleccionado por el usuario, con el nombre que este quiera.
Necesidad	Alta.
Prioridad	Básica
Fuente	Usuario
Estabilidad	Alta

4.5.1.8. URC08 Mostrar Manual Usuario

URC08	Manual usuario
Descripción	El sistema será capaz de mostrar el manual de usuario que estará disponible en formato PDF, a petición del usuario.
Necesidad	Alta
Prioridad	Básica
Fuente	Usuario
Estabilidad	Alta

4.5.2. Requisitos de restricción

4.5.2.1. URR01 Sistema Operativo

URR01	Sistema Operativo
Descripción	El entorno donde será desarrollada y mantenida la aplicación, será Windows XP o cualquier sistema operativo compatible con el mismo y su versión
Necesidad	Alta
Prioridad	Básica
Fuente	Usuario
Estabilidad	Alta

4.5.2.2. URR02 Resolución

URR02	Resolución
Descripción	El sistema debe estar diseñado para ser ejecutado bajo una resolución mínima de 800*600 píxeles.
Necesidad	Alta
Prioridad	Básica
Fuente	Diseñador
Estabilidad	Alta

4.5.2.3. URR03 Mensajes de error

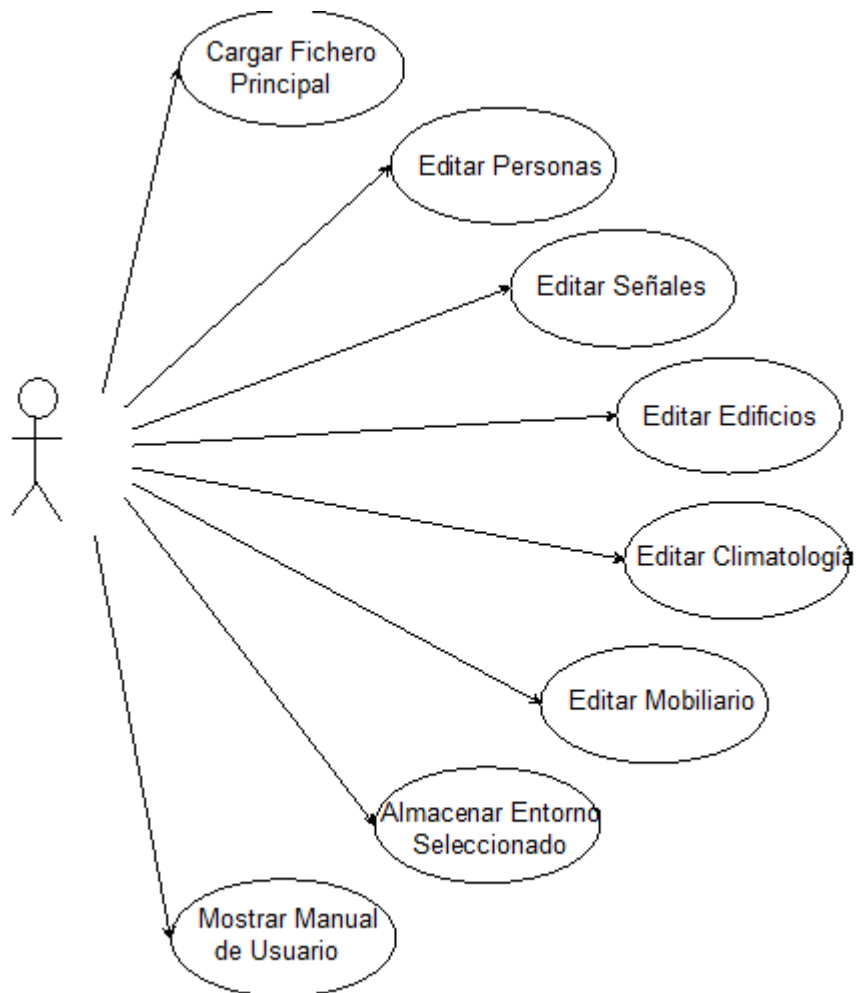
URR03	Mensajes de error
Descripción	En caso de cargar mal el fichero principal, el sistema se lo notificará al usuario mediante un mensaje.
Necesidad	Alta
Prioridad	Básica
Fuente	Diseñador
Estabilidad	Alta

4.5.2.4. URR04 Idioma

URR04	Idioma
Descripción	El idioma en el que se mostrará al usuario toda la información relativa al sistema será el castellano.
Necesidad	Alta
Prioridad	Básica
Fuente	Usuario
Estabilidad	Alta

4.6. CASOS DE USO.

4.6.1. Diagrama de Casos de Uso General.





4.6.2. CU001 Caso de Uso Cargar Fichero Principal.

CU001	
Nombre	Cargar Fichero Principal
Actores	Usuario
Objetivo	Obtener la dirección completa y el nombre donde está almacenado el entorno completo.
Precondiciones	Tener el entorno almacenado en un archivo XML
Poscondiciones	Ninguna
Escenario	<ol style="list-style-type: none"> 1. El usuario pulsa en botón de Inicio y posteriormente Cargar Fichero. 2. Navega por la ventana de navegación que aparece hasta encontrar el fichero. 3. Lo selecciona y pulsa aceptar.
Trazabilidad	RUC001

4.6.3. CU002 Caso de Uso Editar Personas.

CU002	
Nombre	Editar Personas
Actores	Usuario
Objetivo	Que el usuario pueda seleccionar los tipos de personas que desea para su entorno, así como la ubicación de algunas de ellas.
Precondiciones	Tener el entorno cargado, es decir que haya ocurrido el caso de uso CU001.
Poscondiciones	Ninguna
Escenario	<ol style="list-style-type: none"> 1. El usuario selecciona las personas que quiere que aparezcan en su entorno virtual. Podrá elegir entre: <ol style="list-style-type: none"> a. Personas haciendo deporte. b. Personas en grupo. c. Personas cruzando la carretera. d. Número de estudiantes dentro del recinto del campus. 2. Una vez seleccionados pulsará en otra pestaña para seleccionar otras opciones del entorno. Podrá volver a esta pestaña en cualquier momento.
Trazabilidad	RUC002

**4.6.4. CU003 Caso de Uso Editar Señales.**

CU003	
Nombre	Editar Personas
Actores	Usuario
Objetivo	Que el usuario pueda seleccionar los tipos de señales que desea para su entorno.
Precondiciones	Tener el entorno cargado, es decir que haya ocurrido el caso de uso CU001.
Poscondiciones	Ninguna
Escenario	<ol style="list-style-type: none">1. El usuario selecciona las señales que quiere que aparezcan en su entorno virtual. Podrá elegir entre:<ol style="list-style-type: none">a. Señales horizontales (Marcadas en el asfalto).b. Señales Verticales:<ul style="list-style-type: none">• De prioridad y prohibición de entrada (Ceda el paso, Stop,...).• De prohibición (Velocidad).• Indicaciones generales (Paso de peatones).2. Una vez seleccionados pulsará en otra pestaña para seleccionar otras opciones del entorno. Podrá volver a esta pestaña en cualquier momento.
Trazabilidad	RUC003

**4.6.5. CU004 Caso de Uso Editar Edificios.**

CU004	
Nombre	Editar Edificios
Actores	Usuario
Objetivo	Que el usuario pueda seleccionar los edificios que desea para su entorno.
Precondiciones	Tener el entorno cargado, es decir que haya ocurrido el caso de uso CU001.
Poscondiciones	Ninguna
Escenario	<ol style="list-style-type: none">1. El usuario selecciona los edificios que quiere que aparezcan en su entorno virtual. Podrá elegir entre:<ol style="list-style-type: none">a. Edificio Padre Soler.b. Edificio Betancourt.c. Edificio Rey Pastor.d. Edificio Sabatini.e. Edificio Torres Quevedo.f. Polideportivo Alfredo Di Stefano.g. Caldera.2. Una vez seleccionados pulsará en otra pestaña para seleccionar otras opciones del entorno. Podrá volver a esta pestaña en cualquier momento.
Trazabilidad	RUC004

**4.6.6. CU005 Caso de Uso Editar Climatología.**

CU005	
Nombre	Editar Climatología
Actores	Usuario
Objetivo	Que el usuario pueda seleccionar la climatología y el momento del día que desea para su entorno.
Precondiciones	Tener el entorno cargado, es decir que haya ocurrido el caso de uso CU001.
Poscondiciones	Ninguna
Escenario	<ol style="list-style-type: none">1. El usuario selecciona la climatología que quiere en su entorno virtual. Podrá elegir entre:<ol style="list-style-type: none">a. Soleado.b. Nublado.c. Lluvioso.2. El usuario selecciona el momento del día que quiere en su entorno virtual. Podrá elegir entre:<ol style="list-style-type: none">a. Mañana.b. Tarde.c. Noche.3. Una vez seleccionados pulsará en otra pestaña para seleccionar otras opciones del entorno. Podrá volver a esta pestaña en cualquier momento.
Trazabilidad	RUC005



4.6.7. CU006 Caso de Uso Editar Mobiliario.

CU006	
Nombre	Editar Mobiliario
Actores	Usuario
Objetivo	Que el usuario pueda seleccionar el mobiliario que desea para su entorno.
Precondiciones	Tener el entorno cargado, es decir que haya ocurrido el caso de uso CU001.
Poscondiciones	Ninguna
Escenario	<ol style="list-style-type: none"> 1. El usuario selecciona el mobiliario que quiere en su entorno virtual. Podrá elegir entre: <ol style="list-style-type: none"> a. Bancos. b. Vegetación. 2. Una vez seleccionados pulsará en otra pestaña para seleccionar otras opciones del entorno. Podrá volver a esta pestaña en cualquier momento.
Trazabilidad	RUC006

4.6.8. CU007 Caso de Uso Almacenar Entorno Seleccionado.

CU007	
Nombre	Almacenar Entorno Seleccionado
Actores	Usuario
Objetivo	Que el usuario almacene el entorno con las características que ha seleccionado en la misma dirección donde se encuentra el archivo principal.
Precondiciones	Tener el entorno cargado, es decir que haya ocurrido el caso de uso CU001.
Poscondiciones	Ninguna
Escenario	<ol style="list-style-type: none"> 1. El usuario pulsa en botón de Inicio y posteriormente Guardar Fichero. 2. Escribe el nombre que desea para su entorno. 3. Pulsa aceptar.
Trazabilidad	RUC007

4.6.9. CU008 Caso de Uso Mostrar Manual de Usuario.

CU008	
Nombre	Mostrar Manual de Usuario
Actores	Usuario
Objetivo	Que el usuario sea capaz de visualizar el manual de usuario del que dispone el software, para poder solucionar posibles dudas.
Precondiciones	Ninguna
Poscondiciones	Ninguna
Escenario	<ol style="list-style-type: none">1. El usuario pulsa en botón de Ayuda y posteriormente Manual de Usuario.2. Después de visualizarlo cerrará la ventana.
Trazabilidad	RUC008

4.7. *DIAGRAMAS DE SECUENCIA.*

El diagrama de secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema.

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase.

Mientras que el diagrama de casos de uso permite el modelado de una vista del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes intercambiados entre los objetos.

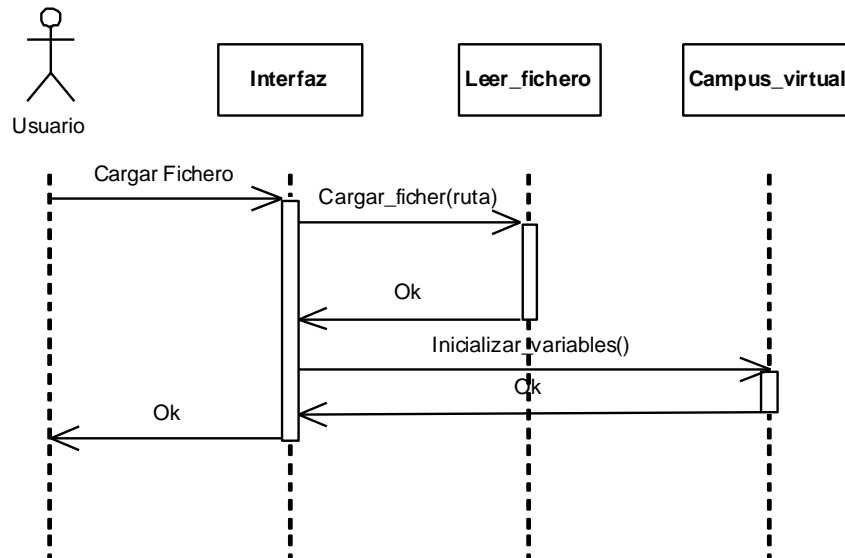
Típicamente uno examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si tienes modelada la descripción de cada caso de uso como una secuencia de varios pasos, entonces puedes "caminar sobre" esos pasos para descubrir qué objetos son necesarios para que se puedan seguir los pasos.

Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como flechas horizontales.

Los mensajes se dibujan cronológicamente desde la parte superior del diagrama a la parte inferior; la distribución horizontal de los objetos es arbitraria.

Durante el análisis inicial, el modelador típicamente coloca el nombre de un mensaje en la línea del mensaje. Más tarde, durante el diseño, el nombre es reemplazado con el nombre del método que está siendo llamado por un objeto en el otro. El método llamado, o invocado, pertenece a la definición de la clase instanciada por el objeto en la recepción final del mensaje.

4.7.1. Diagrama de secuencia DS001 para el caso de uso CU001.

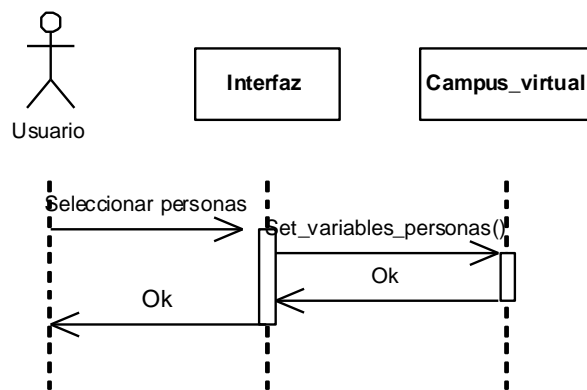


Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso CU001 y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un usuario carga el fichero principal.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El usuario pulsa el botón “Cargar Fichero” en la Interfaz.
- 2) La interfaz llama al método `Cargar_fichero()`, del objeto `Leer_fichero`.
- 3) Este le devuelve un mensaje diciendo que ha cargado correctamente el fichero y se muestra la ruta en la interfaz.
- 4) Una vez obtenido el mensaje de que lo ha cargado correctamente, llama al método `Inicializar_variables()`, para poner las variables del objeto `Campus_virtual` a su valor inicial.

4.7.2. Diagrama de secuencia DS002 para el caso de uso CU002.

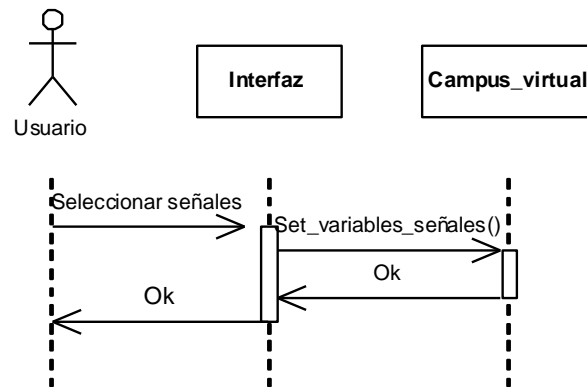


Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso CU002 y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un usuario selecciona las personas que van a aparecer en el entorno virtual.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El usuario selecciona las personas en la Interfaz.
- 2) La interfaz llama al método `Set_variables_personas()` pasándole por parámetros las variables a modificar del objeto `Campus_virtual`.
- 3) Este le devuelve un mensaje diciendo que ha modificado las variables correctamente.

4.7.3. Diagrama de secuencia DS003 para el caso de uso CU003.

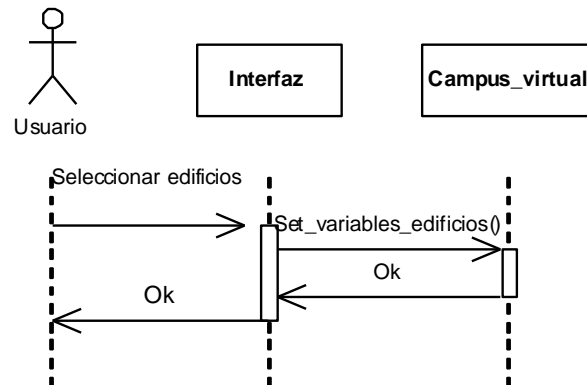


Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso CU003 y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un usuario selecciona las señales que van a aparecer en el entorno virtual.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El usuario selecciona las señales en la Interfaz.
- 2) La interfaz llama al método `Set_variables_señales()` pasándole por parámetros las variables a modificar del objeto `Campus_virtual`.
- 3) Este le devuelve un mensaje diciendo que ha modificado las variables correctamente.

4.7.4. Diagrama de secuencia DS004 para el caso de uso CU004.

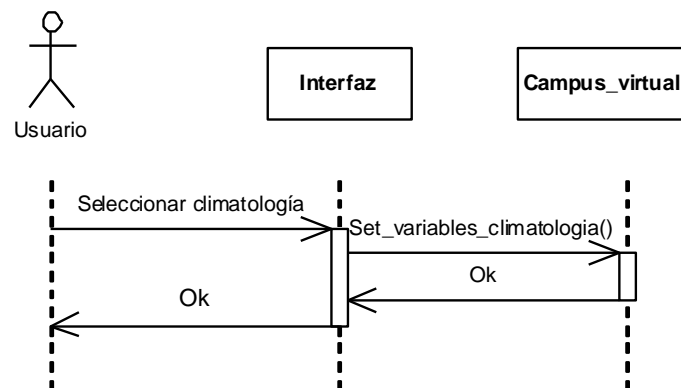


Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso CU004 y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un usuario selecciona los edificios que van a aparecer en el entorno virtual.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El usuario selecciona los edificios en la Interfaz.
- 2) La interfaz llama al método `Set_variables_edificios()` pasándole por parámetros las variables a modificar del objeto `Campus_virtual`.
- 3) Este le devuelve un mensaje diciendo que ha modificado las variables correctamente.

4.7.5. Diagrama de secuencia DS005 para el caso de uso CU005.

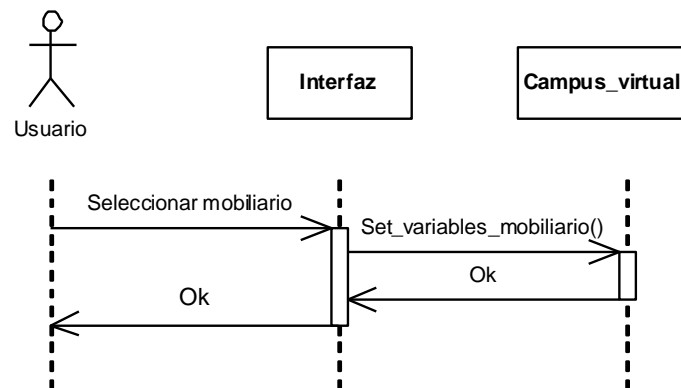


Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso CU005 y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un usuario selecciona la climatología y el tiempo que va a aparecer en el entorno virtual.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El usuario selecciona la climatología y la hora en la Interfaz.
- 2) La interfaz llama al método `Set_variables_climatología()` pasándole por parámetros las variables a modificar del objeto `Campus_virtual`.
- 3) Este le devuelve un mensaje diciendo que ha modificado las variables correctamente.

4.7.6. Diagrama de secuencia DS006 para el caso de uso CU006.

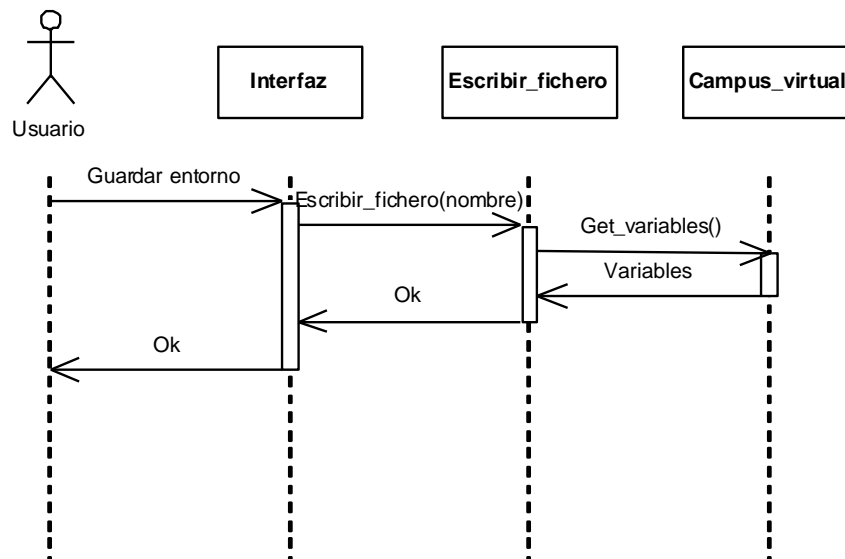


Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso CU006 y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un usuario selecciona el mobiliario que va a aparecer en el entorno virtual.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El usuario selecciona el mobiliario en la Interfaz.
- 2) La interfaz llama al método `Set_variables_mobiliario()` pasándole por parámetros las variables a modificar del objeto `Campus_virtual`.
- 3) Este le devuelve un mensaje diciendo que ha modificado las variables correctamente.

4.7.7. Diagrama de secuencia DS007 para el caso de uso CU007.

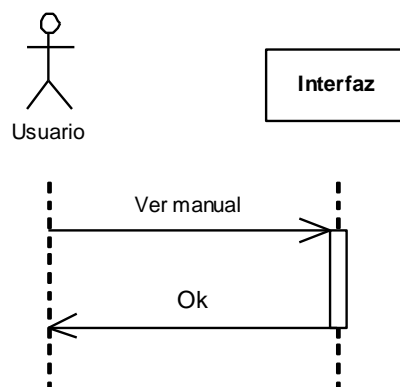


Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso CU007 y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un usuario guarda su entorno deseado.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El usuario escribe el nombre deseado y pulsa “Guardar”.
- 2) La interfaz llama al método `Escribir_fichero()`, del objeto `Leer_fichero`, pasándole por parámetros el nombre del fichero.
- 3) Este a su vez obtiene los valores de las variables, mediante el método `Get_variables`, para escribir el fichero, dependiendo del valor de las mismas.
- 4) Este le devuelve un mensaje diciendo que ha generado correctamente el fichero.

4.7.8. Diagrama de secuencia DS008 para el caso de uso CU008.



Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso CU008 y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un usuario quiere visualizar el manual de usuario.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El usuario pulsa “Mostrar manual” en la interfaz.
- 2) La interfaz se lo muestra al usuario.

4.8. TRAZABILIDAD.

Trazabilidad significa capturar las relaciones de implementación y dependencia en el modelo, siguiendo un histórico de los pasos seguidos para desarrollar el software.

Por ejemplo:

- 1) Un proceso de negocio requerirá estudiarlo para poder obtener del cliente que nos reclama dicho software una serie de “Requisitos”.
- 2) A partir de dichos requisitos definiremos las principales funcionalidades del sistema “Casos de Uso” para implementar las funciones del proceso.
- 3) De estudiar, dividiendo en pasos y tareas estos casos de uso, surgirán más tarde los “Diagramas de secuencia”.

El proceso anterior nos hace ver una serie de pasos que van avanzando desde un lenguaje cercano al nuestro (lenguaje natural) hacia un lenguaje más cercano al software.

En el caso del software desarrollado en este proyecto, un ejemplo de la trazabilidad desde un requisito hasta su diagrama de secuencia correspondiente sería el siguiente:

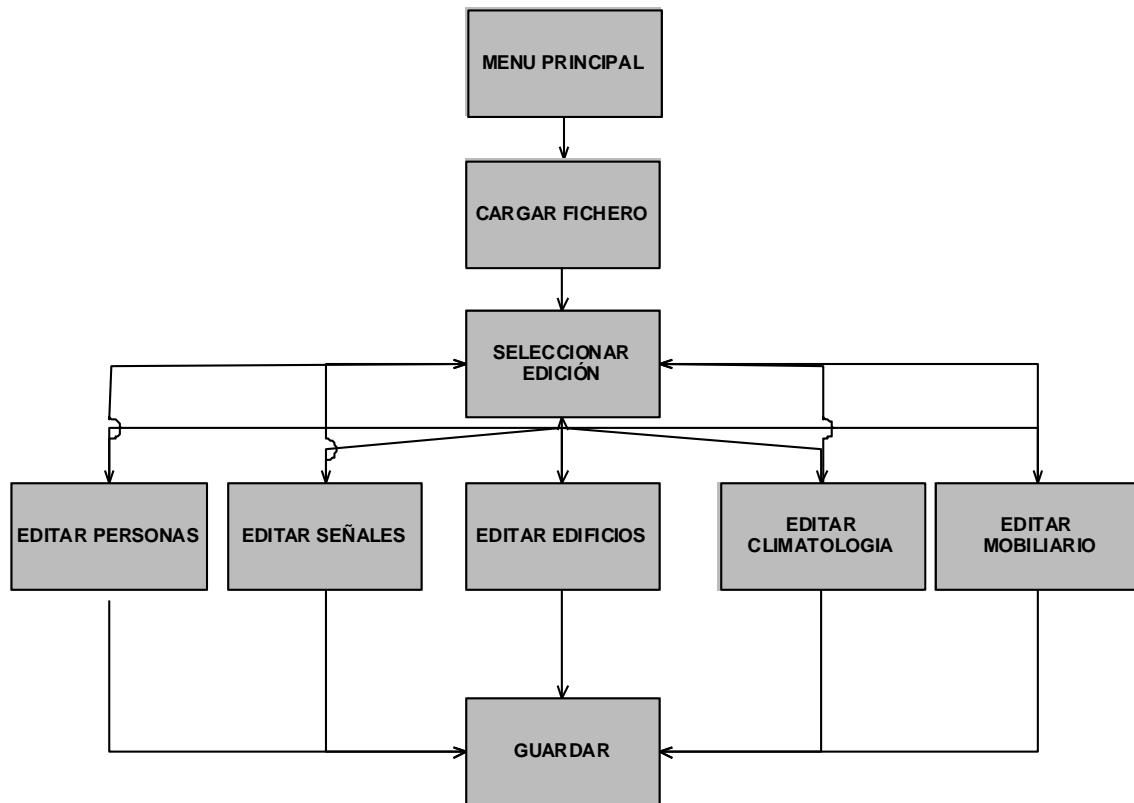
REQUISITO (RUC001) → CASO DE USO (CU001) → DIAGRAMA DE SECUENCIA (DS 001)

4.8.1. Matriz de trazabilidad.

MATRIZ DE TRAZABILIDAD		
REQUISITOS DE CAPACIDAD	CASOS DE USO	DIAGRAMA DE SECUENCIA
RUC001	CU001	DS001
RUC002	CU002	DS002
RUC003	CU003	DS003
RUC004	CU004	DS004
RUC005	CU005	DS005
RUC006	CU006	DS006
RUC007	CU007	DS007
RUC008	CU008	DS008

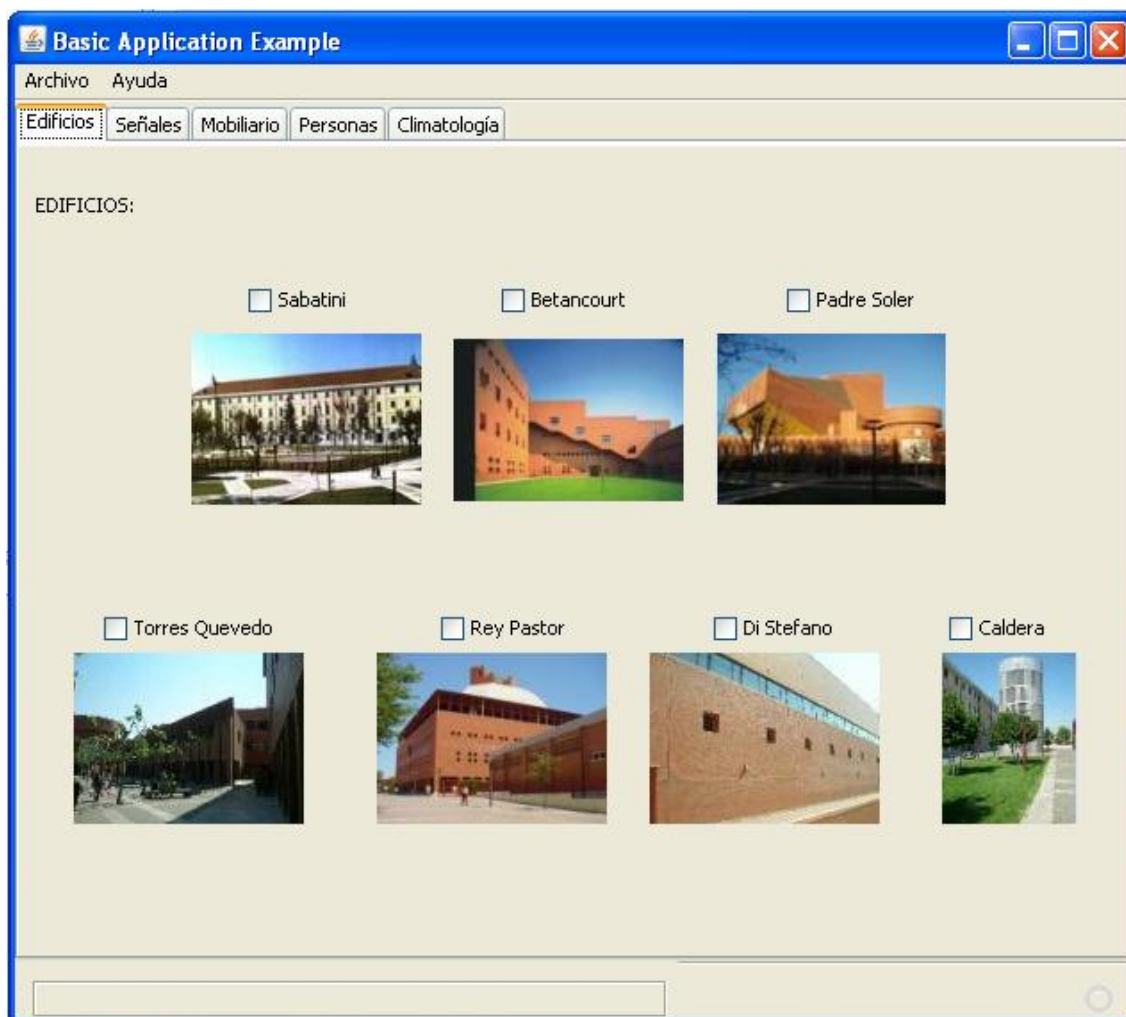
4.9. DISEÑO DE LA HERRAMIENTA.

El siguiente diagrama muestra la jerarquía de diseño planeada para generar nuestro software, desde el punto de vista del usuario.



A continuación, vamos a hacer una descripción gráfica del diseño del software, de modo que se vaya viendo cómo se ha decidido diseñar cada aspecto acorde a los requisitos funcionales especificados.

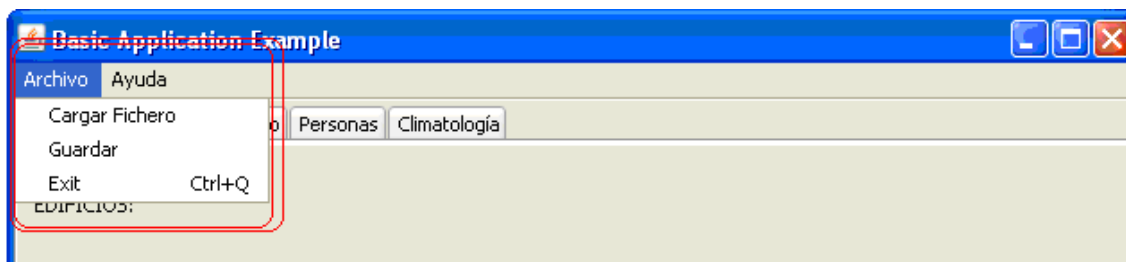
4.9.1. *Diseño del menú principal.*



Como se puede observar en la captura del menú principal, vemos que el software está dividido por zonas, siguiendo el esquema siguiente:

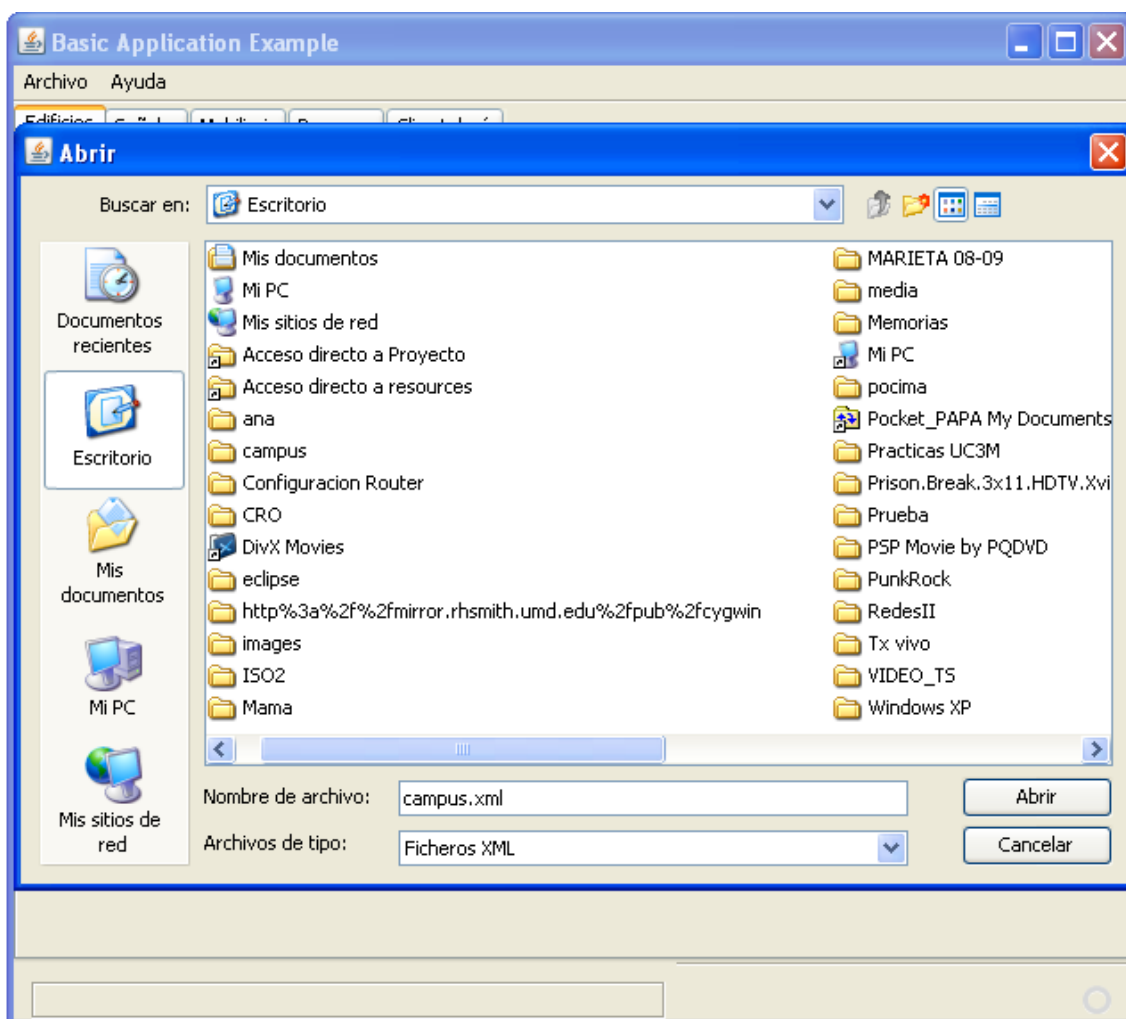
- 1) En la parte superior: una barra herramientas.
- 2) A continuación: una barra de navegación con una pestaña para cada uno de los distintos menús de edición.
- 3) En la parte central: ahí se mostrarán cada uno de los contenidos de los menús de edición.
- 4) En la parte inferior: se mostrará la ruta cuando se haya cargado correctamente el fichero principal.

4.9.2. *Diseño de la barra de herramientas.*

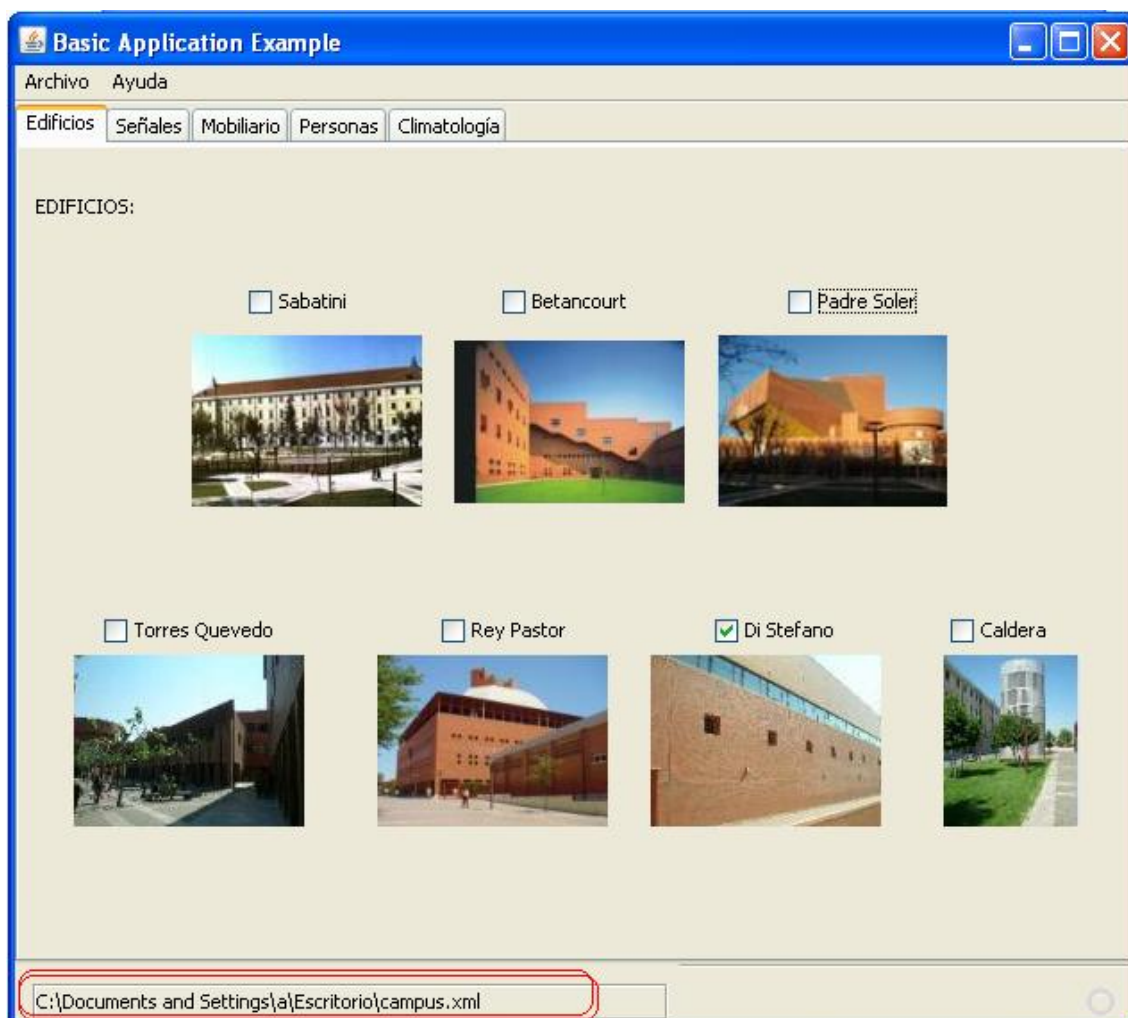


Una vez presionado “Archivo” dentro de la barra de herramientas nos mostrará las opciones de las que disponemos, las cuales son:

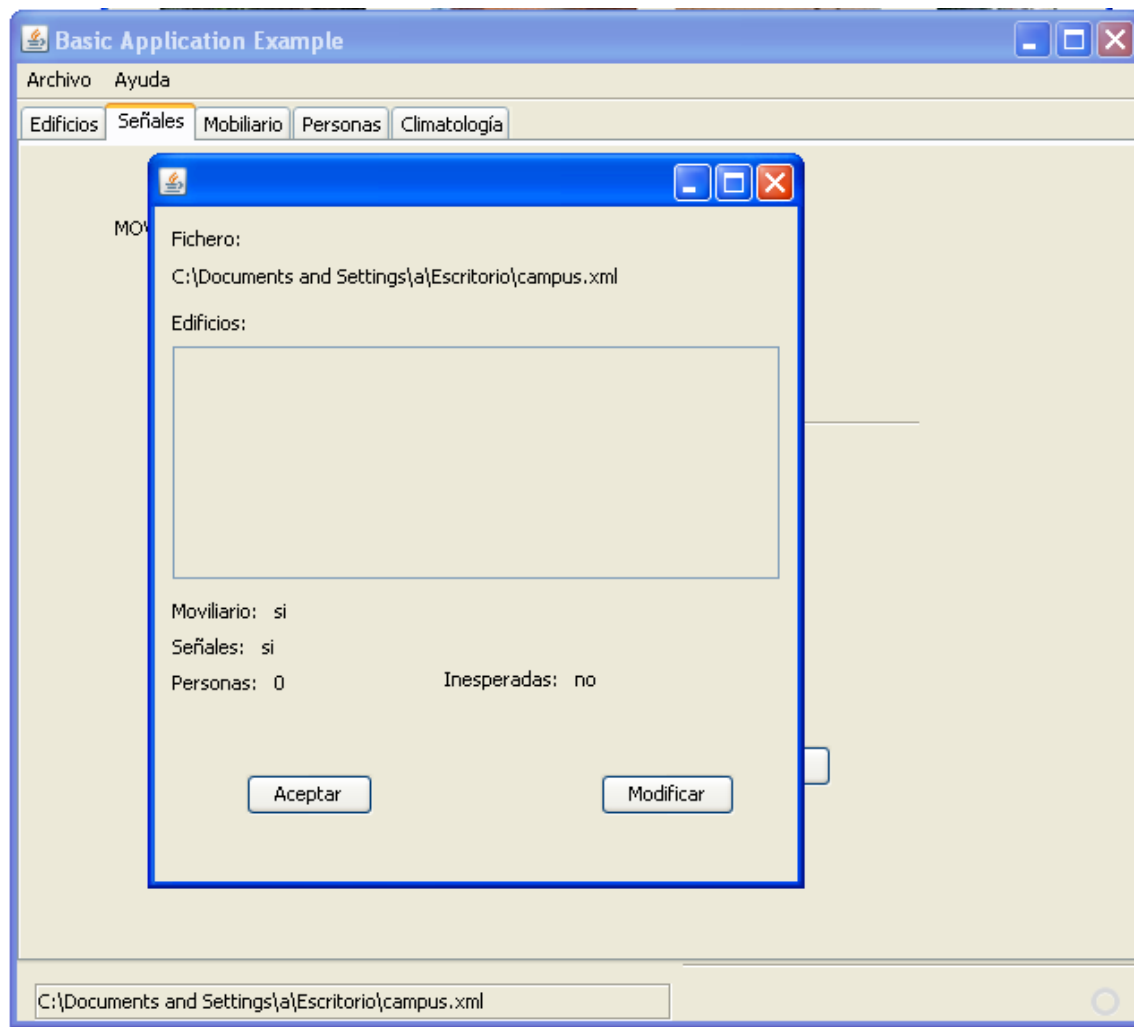
- Cargar Fichero: Una vez pulsada esta opción nos aparecerá un menú de navegación en el que seleccionaremos el archivo principal campus.xml.



Una vez seleccionado este archivo nos mostrará la ruta en el menú principal, esto nos indicará que ya podemos comenzar a realizar las modificaciones en nuestro nuevo entorno.

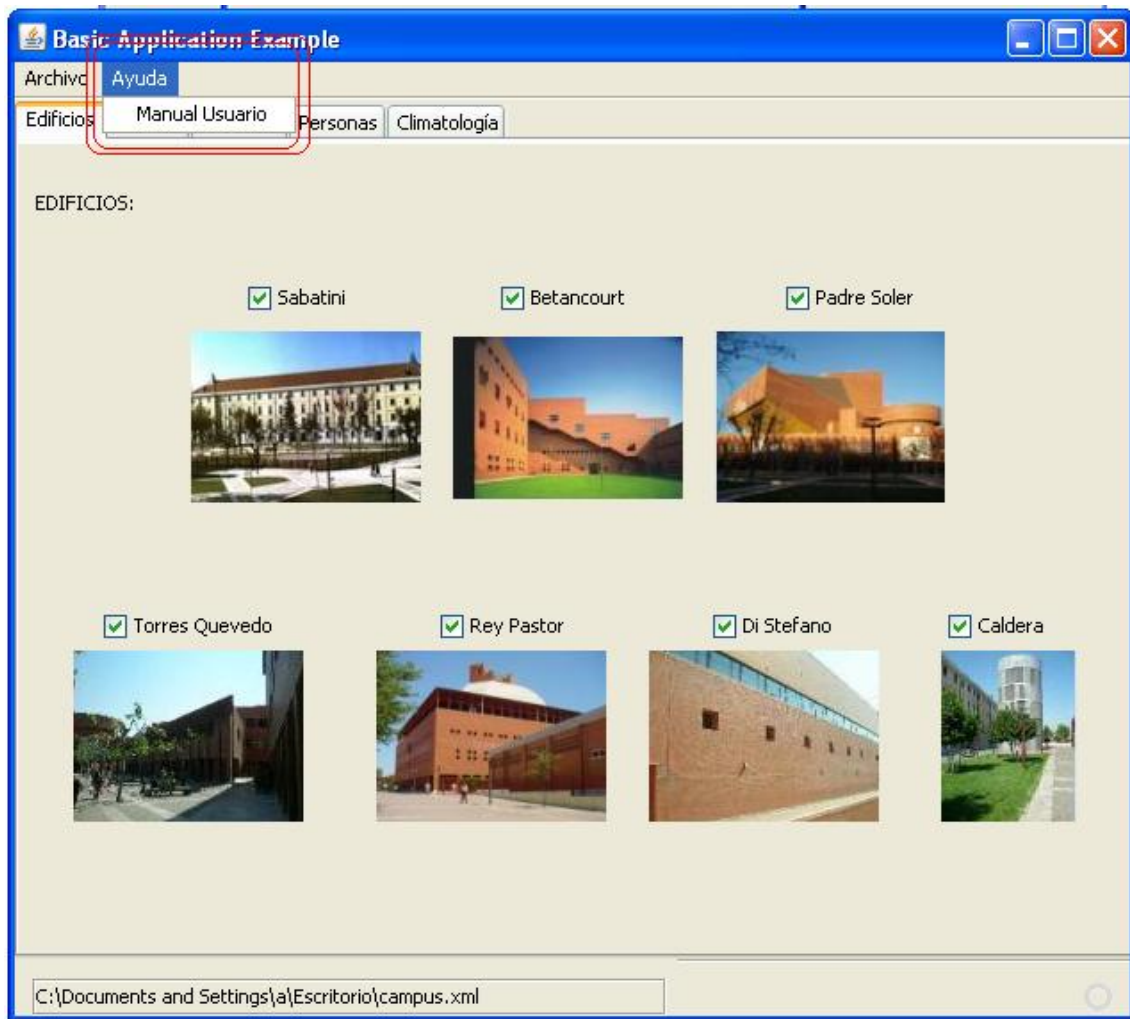


- Guardar: Si pulsamos la opción guardar, el software nos mostrará una pantalla en la que aparecerán todas nuestras preferencias, con lo que podremos confirmarlo si es correcto o modificarlo si nos hemos confundido. Si confirmamos, nos generará nuestro nuevo campus virtual.



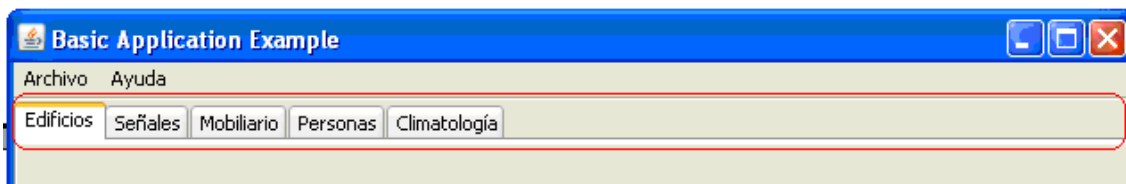
- Salir: Si pulsamos salir, saldrá de nuestra aplicación.

Si en cambio en la barra de herramientas se pulsa ayuda nos dará la opción de consultar el manual de usuario.



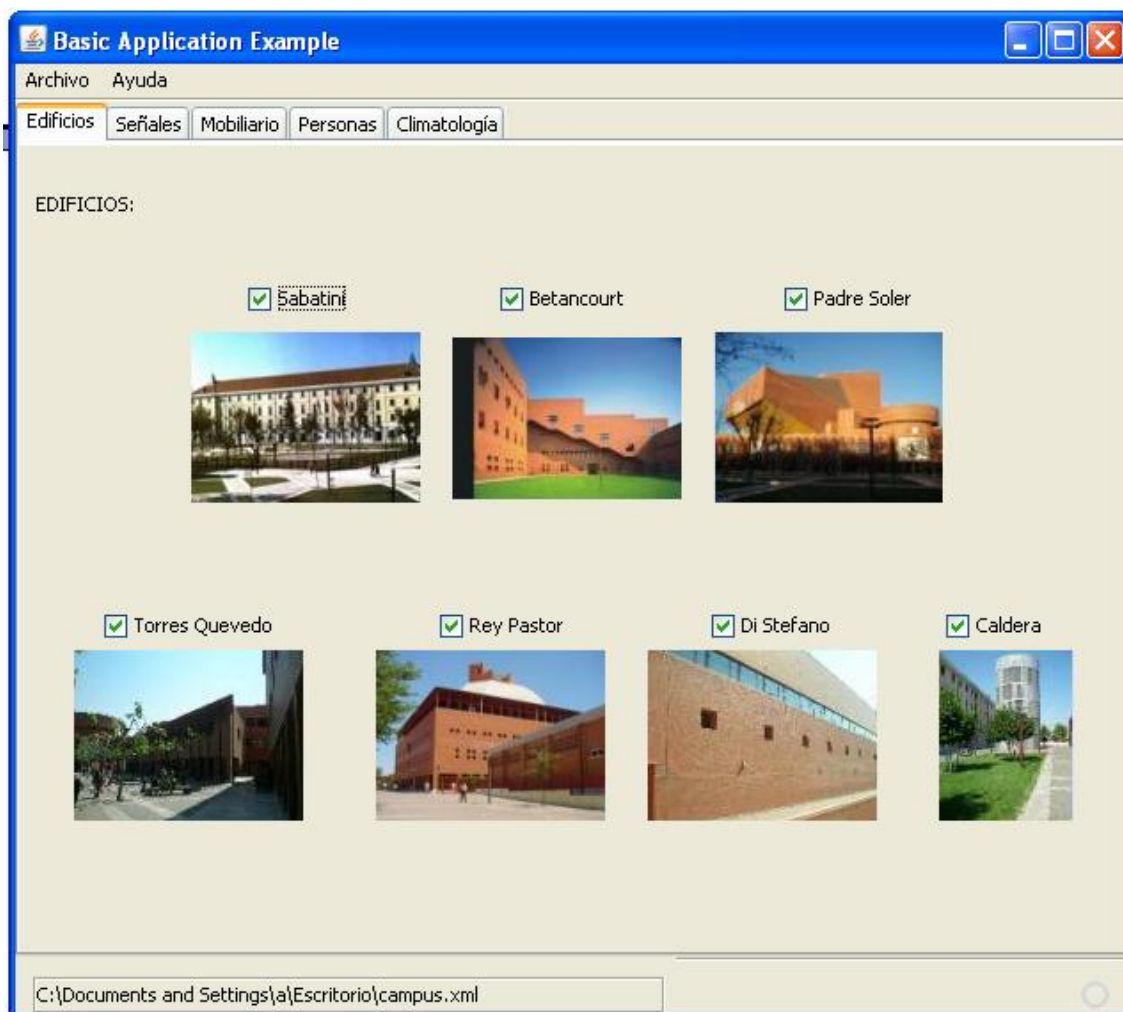
4.9.3. *Diseño de la barra de navegación y menús de edición.*

A continuación se mostrarán cada uno de los menús de edición, a los cuales accederemos desde la barra de navegación.

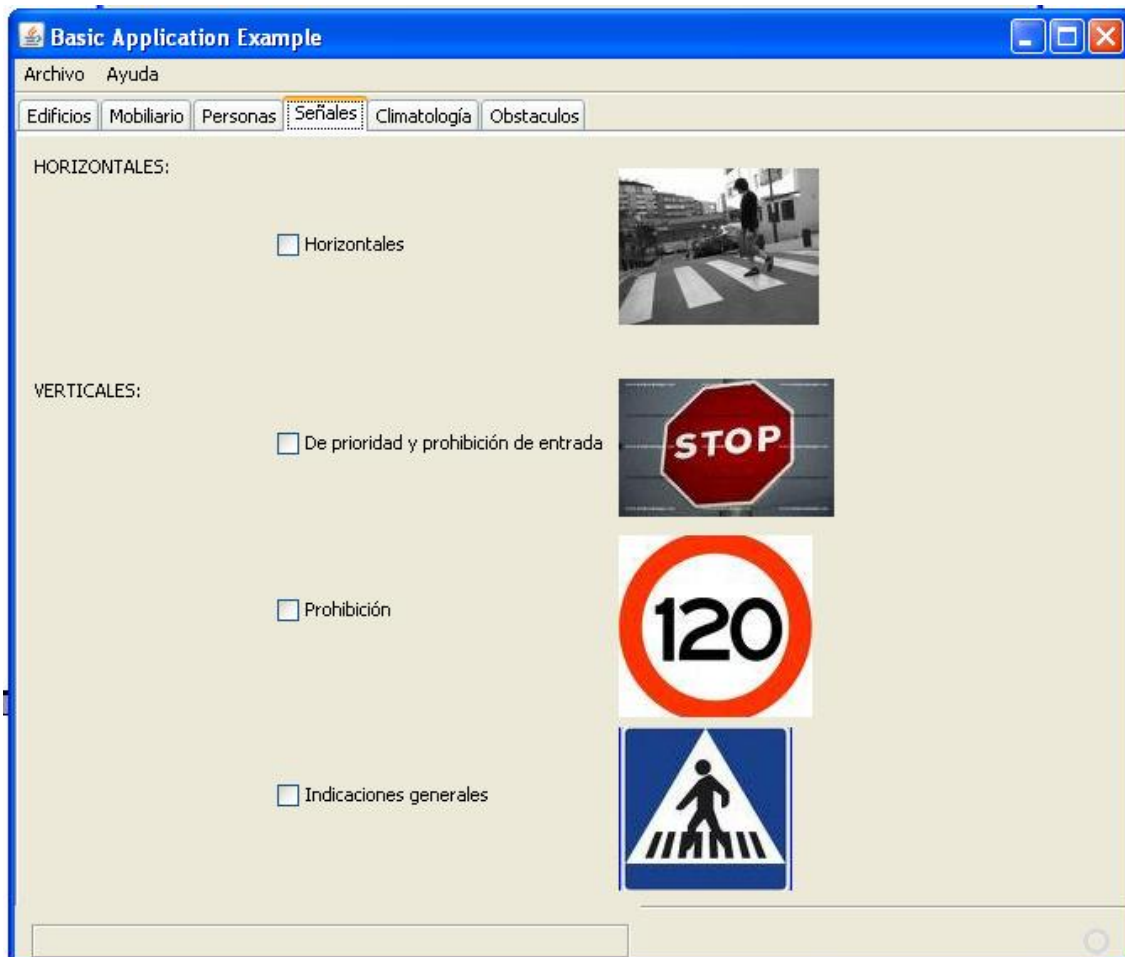


Los posibles menús de edición a los que podemos acceder son los siguientes:

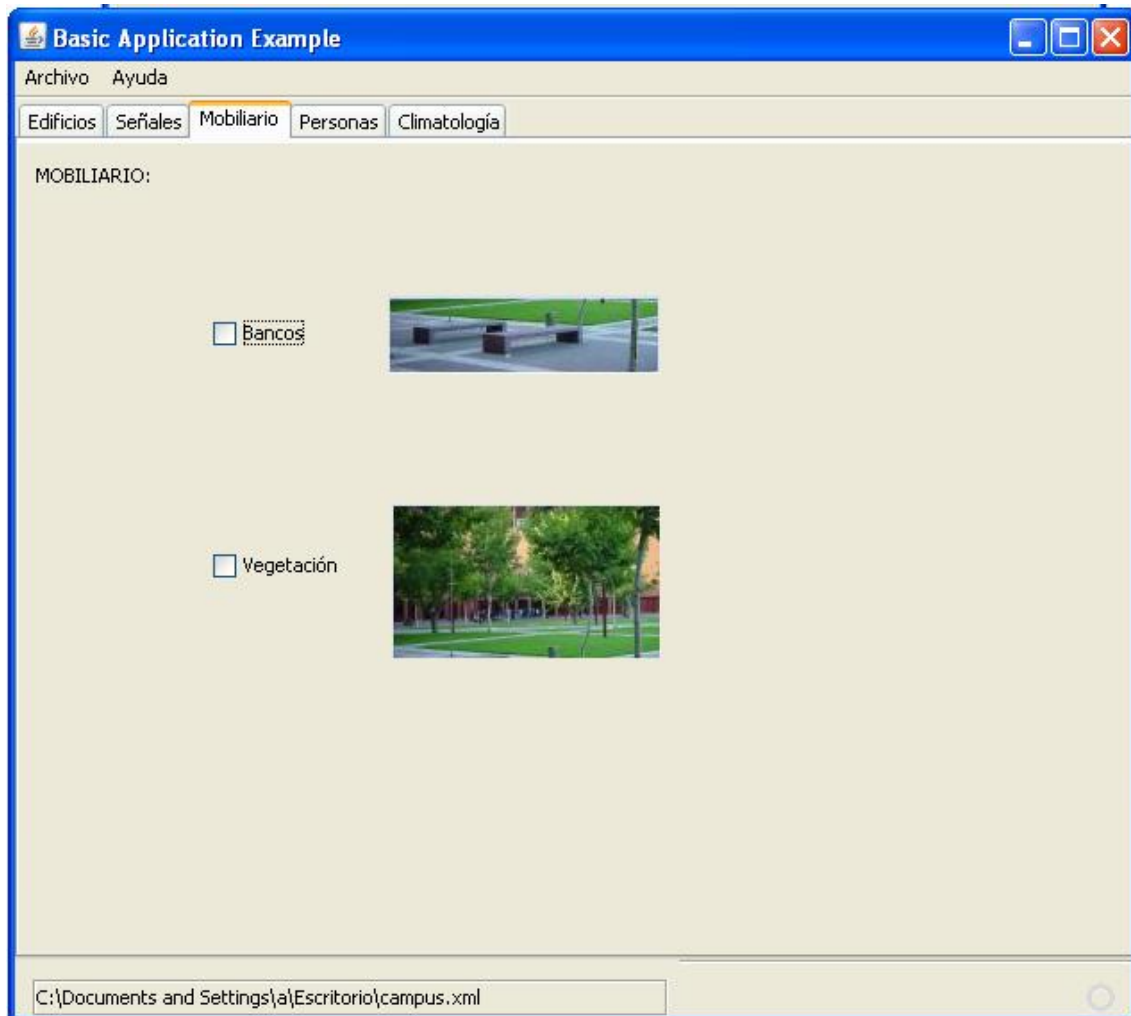
- Edición de Edificios: En este menú se podrán seleccionar los edificios que queremos que formen parte de nuestro campus virtual.



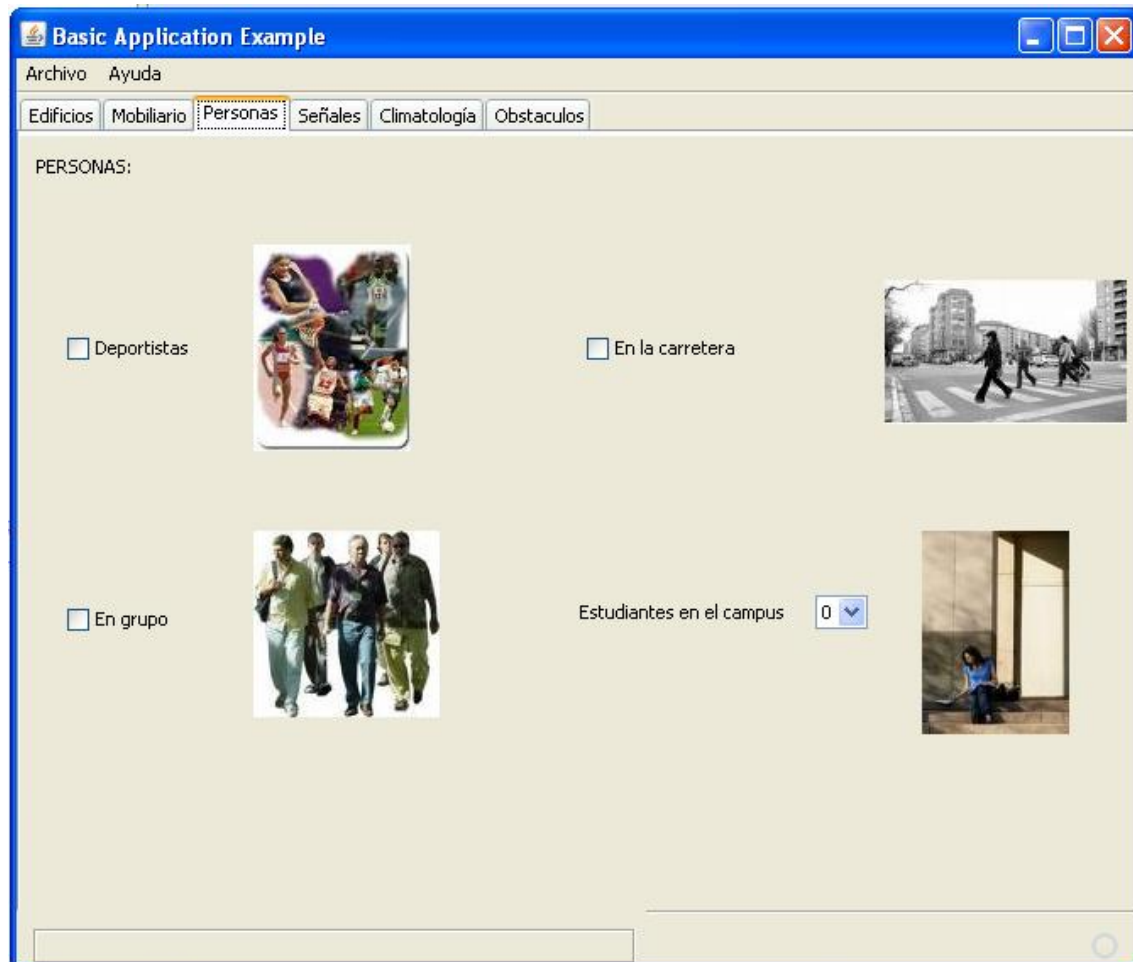
- Edición de Señales: En este menú se podrán seleccionar las señales que queremos que formen parte de nuestro campus virtual.



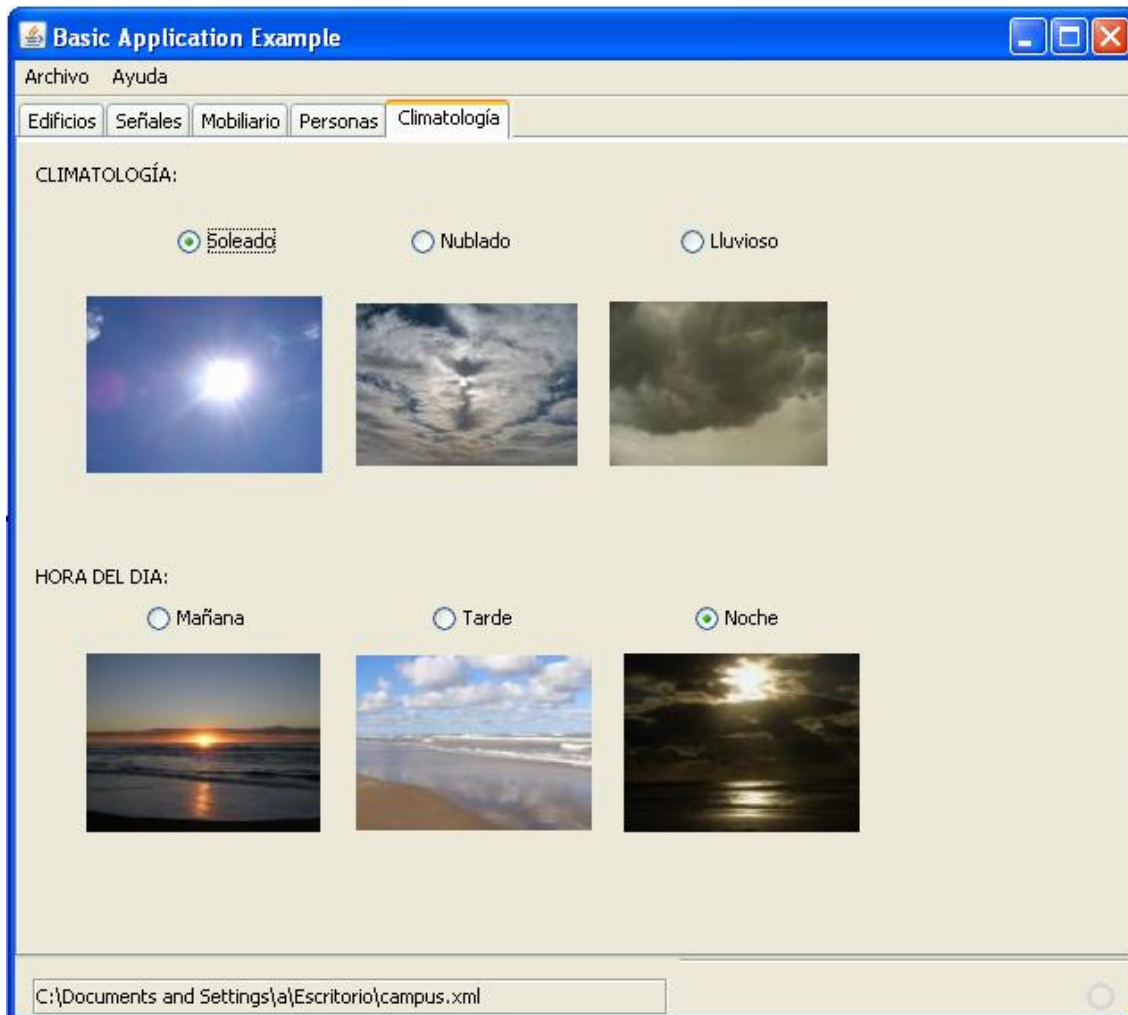
- Edición de Mobiliario: En este menú se podrá seleccionar si se desea que haya o no vegetación y bancos dentro de nuestro campus virtual.



- Edición de Personas: En este menú se podrán seleccionar el tipo de personas que queremos que formen parte de nuestro campus virtual.

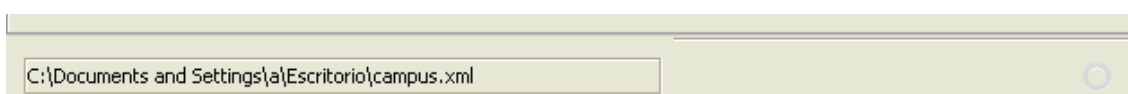


- Edición de Climatología: En este menú se podrá seleccionar el momento del día en el que el usuario quiera que se encuentre su entorno virtual y el tiempo que quiere que haga.



4.9.4. *Diseño del cuadro que almacena la ruta.*

Esta parte es bastante importante, ya que en un principio aparece vacía, lo que nos indica que no se ha cargado el fichero principal. Si se carga correctamente, en ella se nos mostrará continuamente la ruta, indicándonos que podemos proceder a la generación de nuestro nuevo campus virtual.





5. DESCRIPCIÓN DE LOS MODELOS 3D.

En este punto se va a mostrar cada uno de los modelos 3D que he realizado para este proyecto, con el programa Sketchup.

Estos modelos luego se importarán a MRSD de la siguiente manera:

- Primero se genera el modelo en Sketchup, con formato .skp.
- Seguidamente se exporta al formato .obj que si que es reconocido por MRSD.
- Se introduce en MRSD mediante una aplicación desarrollada en C#, en la que le indicamos la posición de cada objeto, así como la posición de la cámara, el tipo de cielo, el tipo de suelo,..
- Se compila, y al ejecutarlo te genera el entorno virtual.

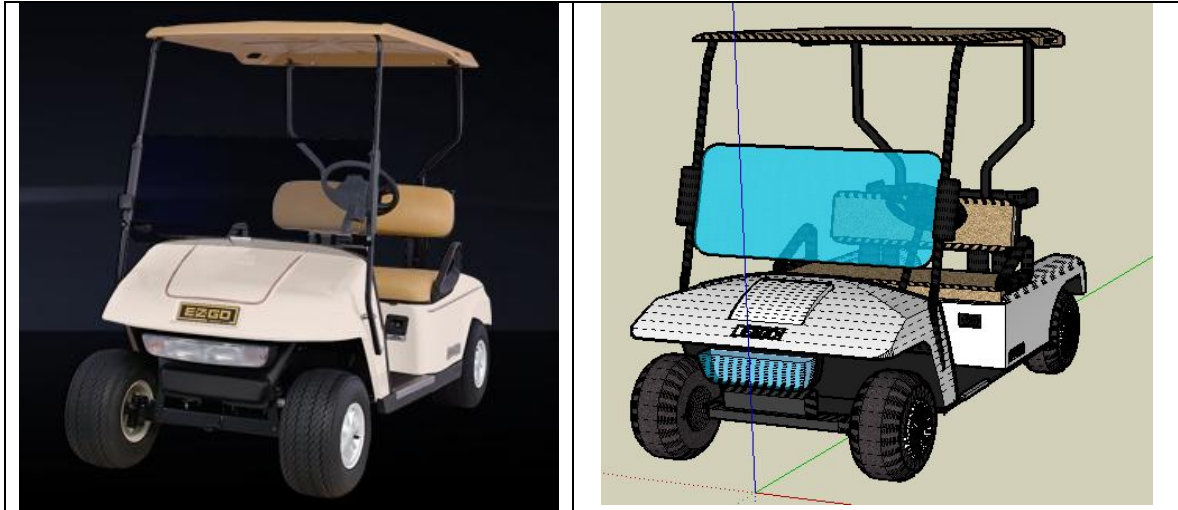
Cada modelo se comparará con una foto real, tratando de buscar el mismo ángulo de la foto.

En el caso de los edificios, también se compararán con una vista sacada desde el Google Earth para que se vea que las dimensiones son semejantes.

Al principio de cada uno, se realizará una breve descripción.

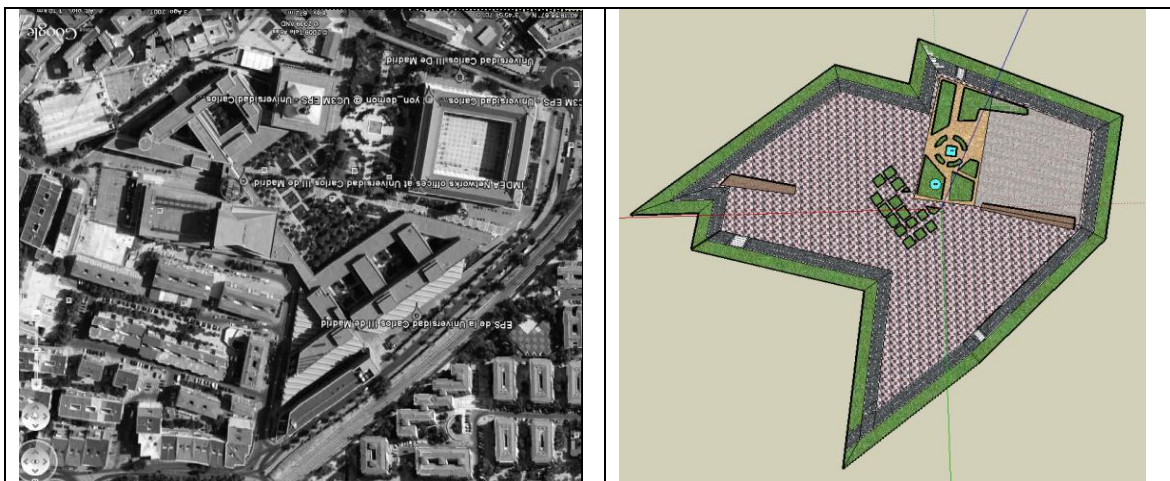
5.1. *Carrito de Golf Freedom SE.*

Carrito de golf fabricado por la compañía EZGO (Textron Company). Todas sus características y especificaciones vienen detalladas en la siguiente dirección web: http://www.ezgo.com/golf/fleet/freedom_se.html



5.2. *Suelo del campus de Leganés.*

Situado en la localidad madrileña de Leganés, al sur de Madrid. En este campus se ubica la Escuela Politécnica Superior. Dotado de unas modernas instalaciones entre las que destaca la biblioteca, salas de ordenadores en varios edificios, laboratorios, zona wifi en todo el campus y un polideportivo con una piscina cubierta, además del Auditorio de la Universidad que acoge todos los años obras de teatro, óperas, conciertos...

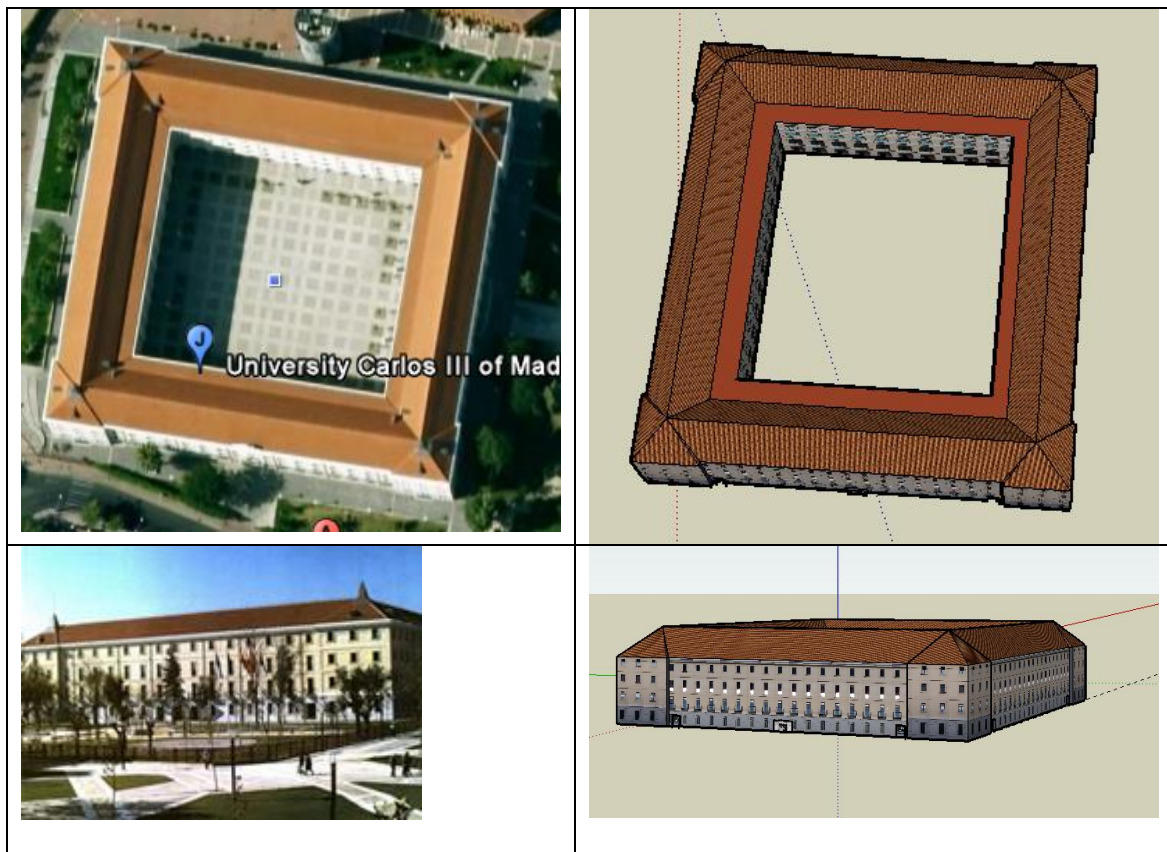


5.3. *Edificio Sabatini.*

Es el edificio número 2 del Campus de Leganés de la universidad Carlos III de Madrid (España), en él se imparten clases de Ingeniería Industrial, Telecomunicaciones e Informática y consta de cuatro plantas, incluida la baja, con una superficie total de 25.474 metros cuadrados.

Este edificio histórico y emblemático para el municipio de Leganés, fue diseñado por Francisco Sabatini en el siglo XVIII, para albergar las guardias Walonas, y desde entonces había tenido un uso militar. La rehabilitación para la docencia ha supuesto una completa modernización de sus instalaciones, pero siempre manteniendo el espíritu original del edificio. Este edificio departamental posee 20 grandes aulas en su planta tercera, cubiertas con bóvedas de madera. El resto de las plantas están ocupadas por aulas informáticas, aulas de seminario, residencia de estudiantes, despachos para profesorado y dirección de la escuela, administración, rectorado, OTRI, contratación y patrimonio, mantenimiento y fundación, contando además con cafetería en la planta baja.

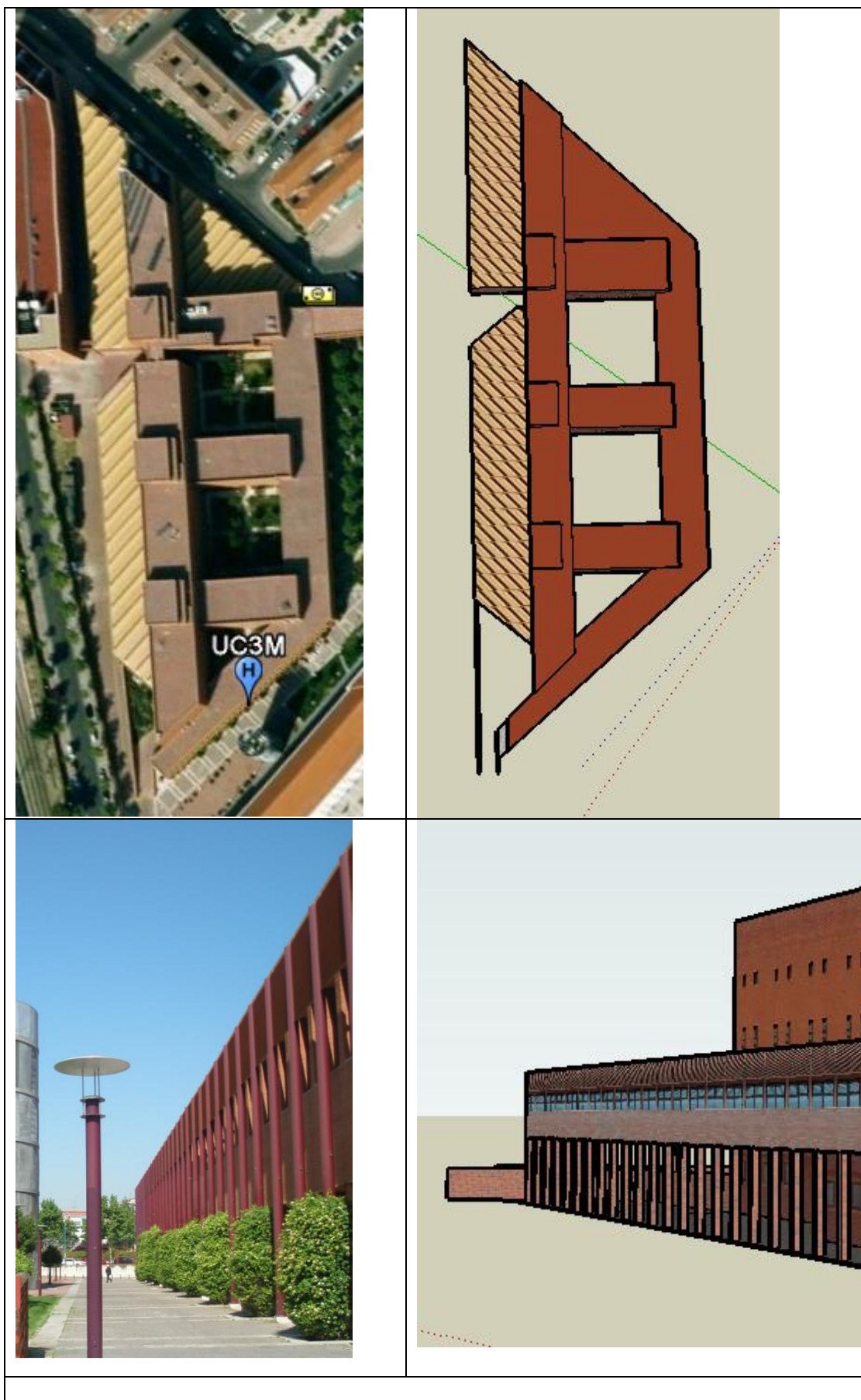
El proyecto de rehabilitación lo redactó el arquitecto Ricardo López de Rego y Uriarte y las obras fueron ejecutadas por la constructora FCC Construcciones de mayo de 1997 a julio de 1998, con un presupuesto total de 1.606.255.456 ptas.

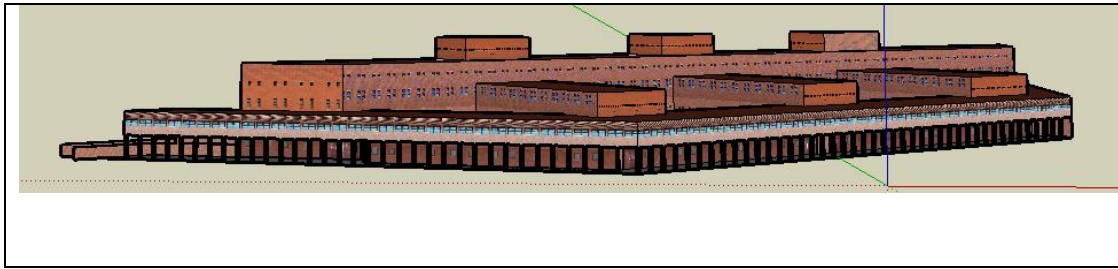


5.4. Edificio Betancourt.

El edificio Betancourt es el número 1 del campus de Leganés, y en su interior alberga lo siguiente:

- Departamentos Universitarios.
- Laboratorios y Talleres.
- Servicio de Informática
- Aulas
- Zona comercial.

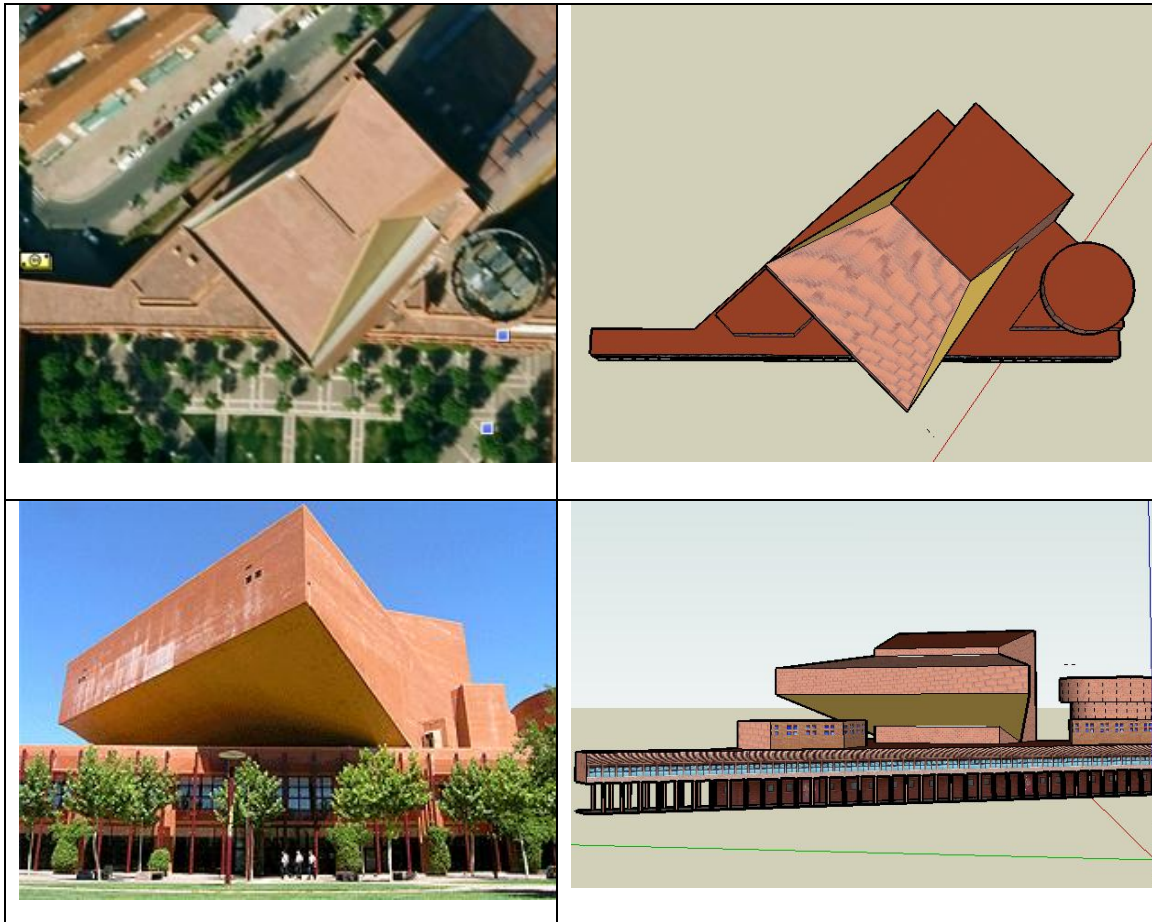




5.5. Edificio Padre Soler.

El edificio Padres Soler es el número 5 del campus de Leganés, y en su interior alberga lo siguiente:

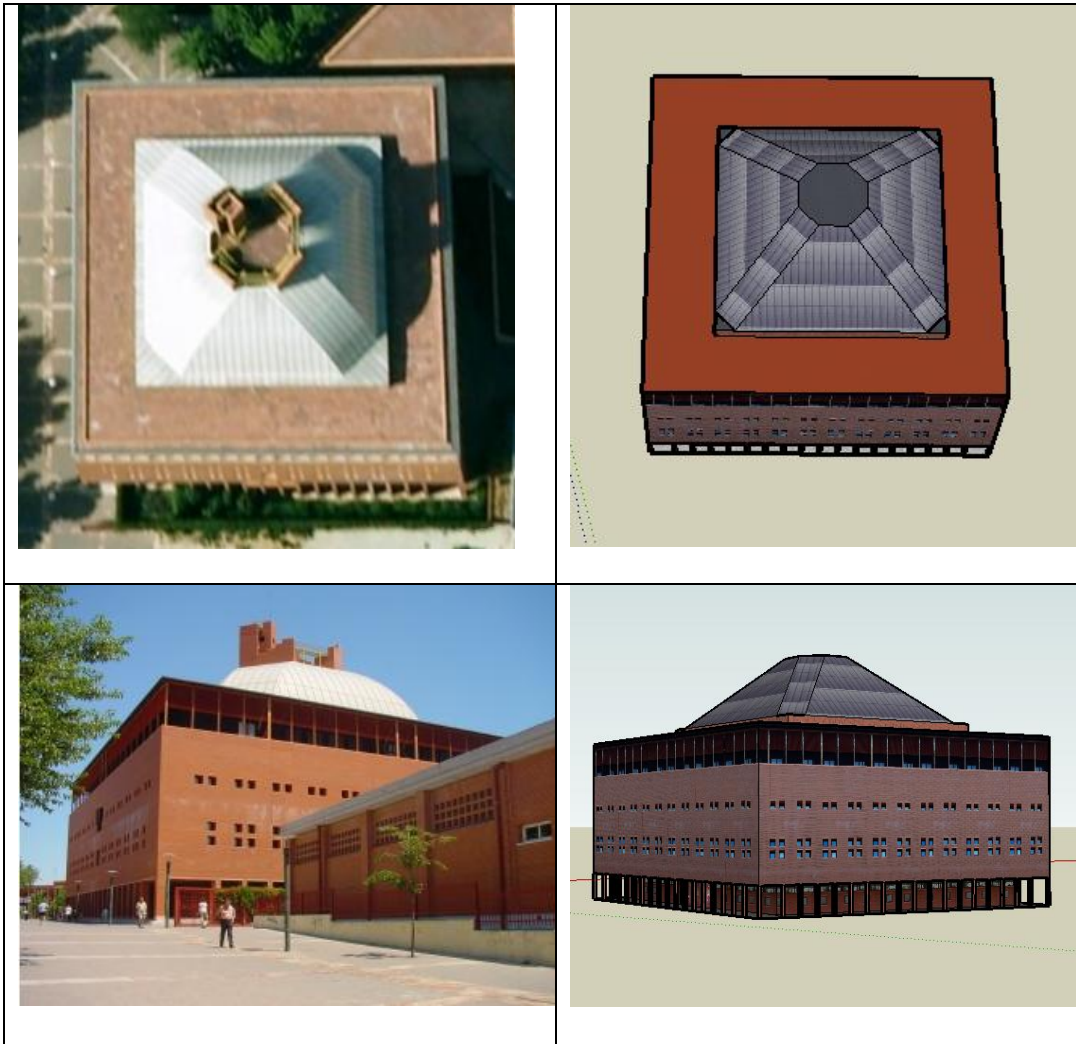
- Auditorio Aula-Magna.
- Aula de Grados.
- Cafetería - Restaurante.



5.6. Edificio Rey Pastor.

El edificio Rey Pastor es el número 3 del campus de Leganés, y en su interior alberga lo siguiente:

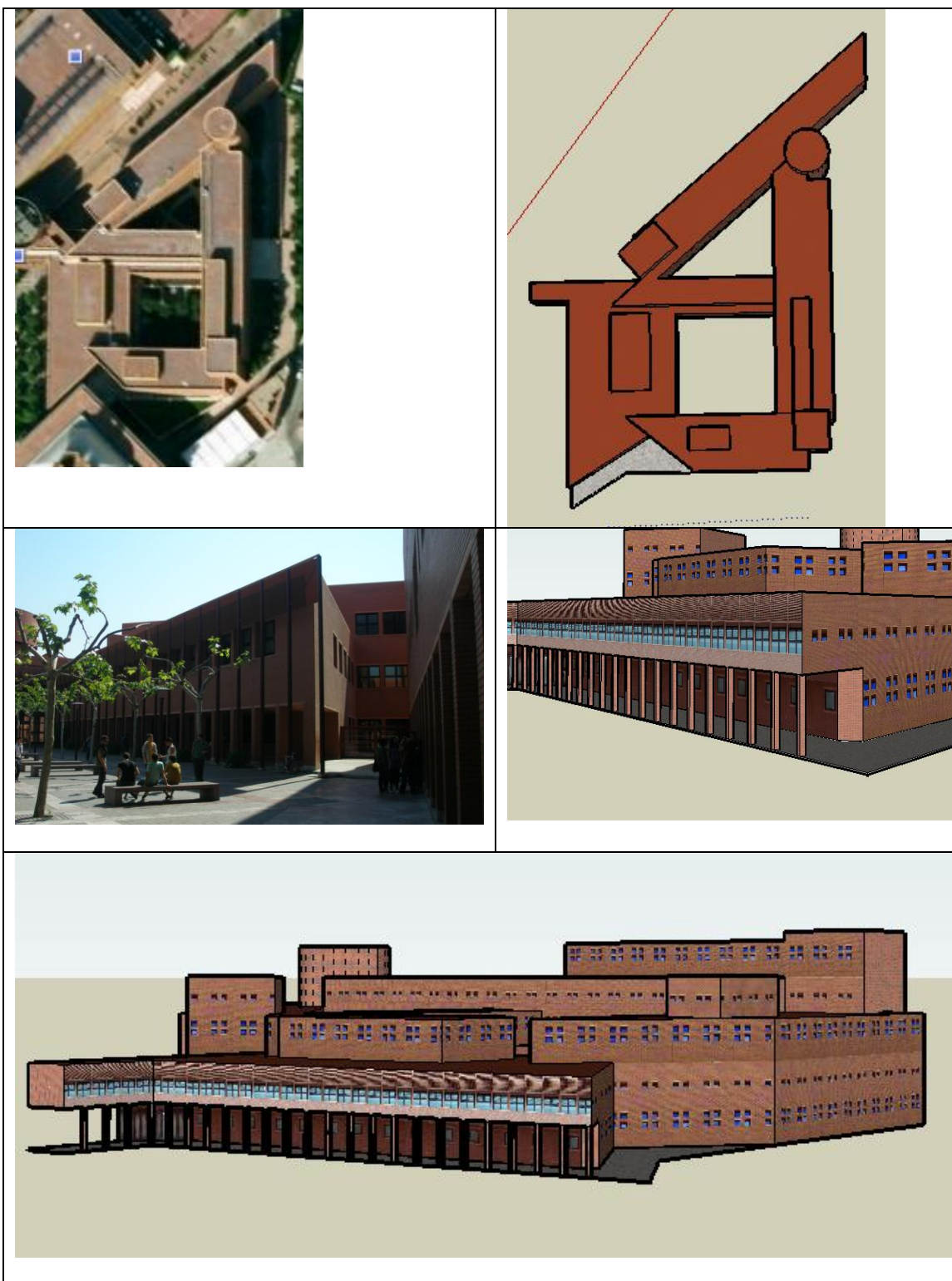
- Biblioteca.
- Archivo General.



5.7. *Edificio Torres Quevedo.*

El edificio Torres Quevedo es el número 4 del campus de Leganés, y en su interior alberga lo siguiente:

- Departamentos Universitarios.
- Aulas.

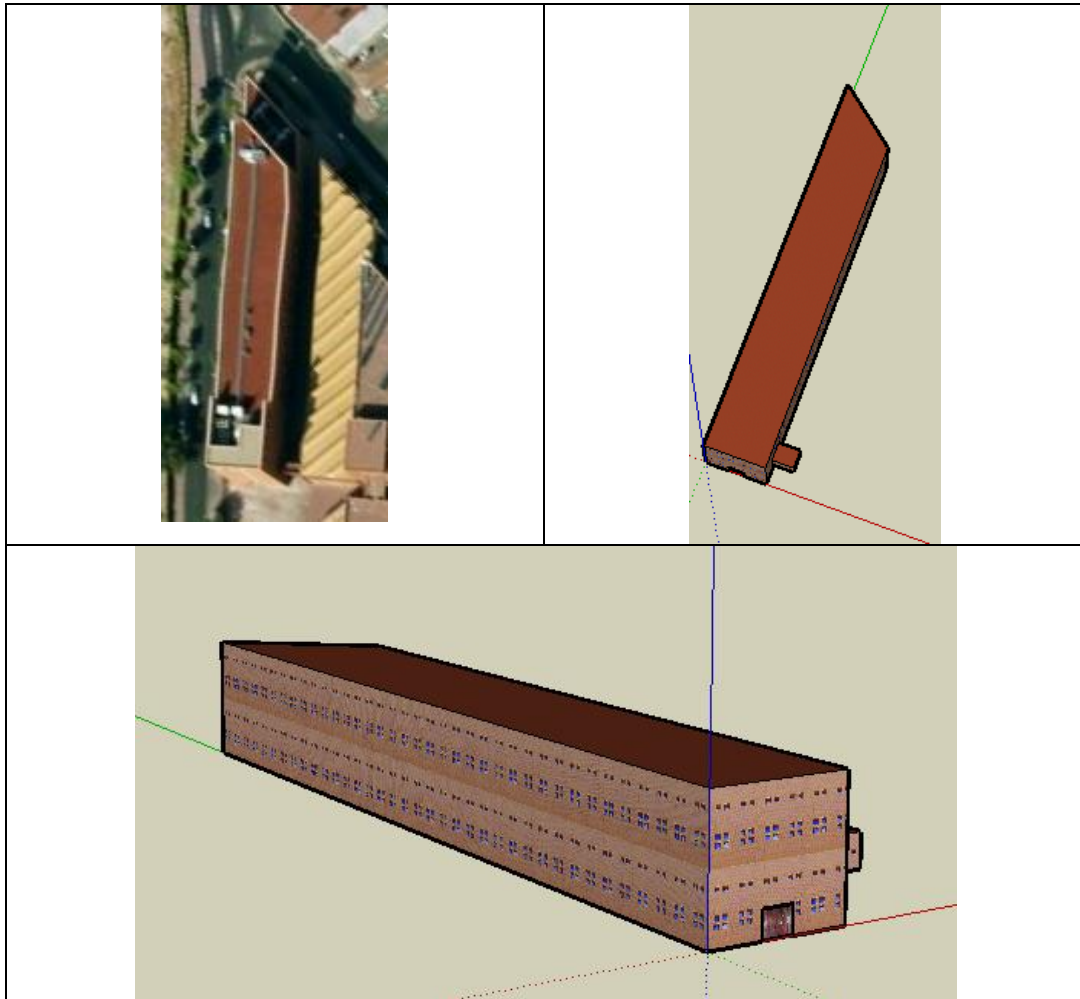




5.8. Edificio Juan Benet.

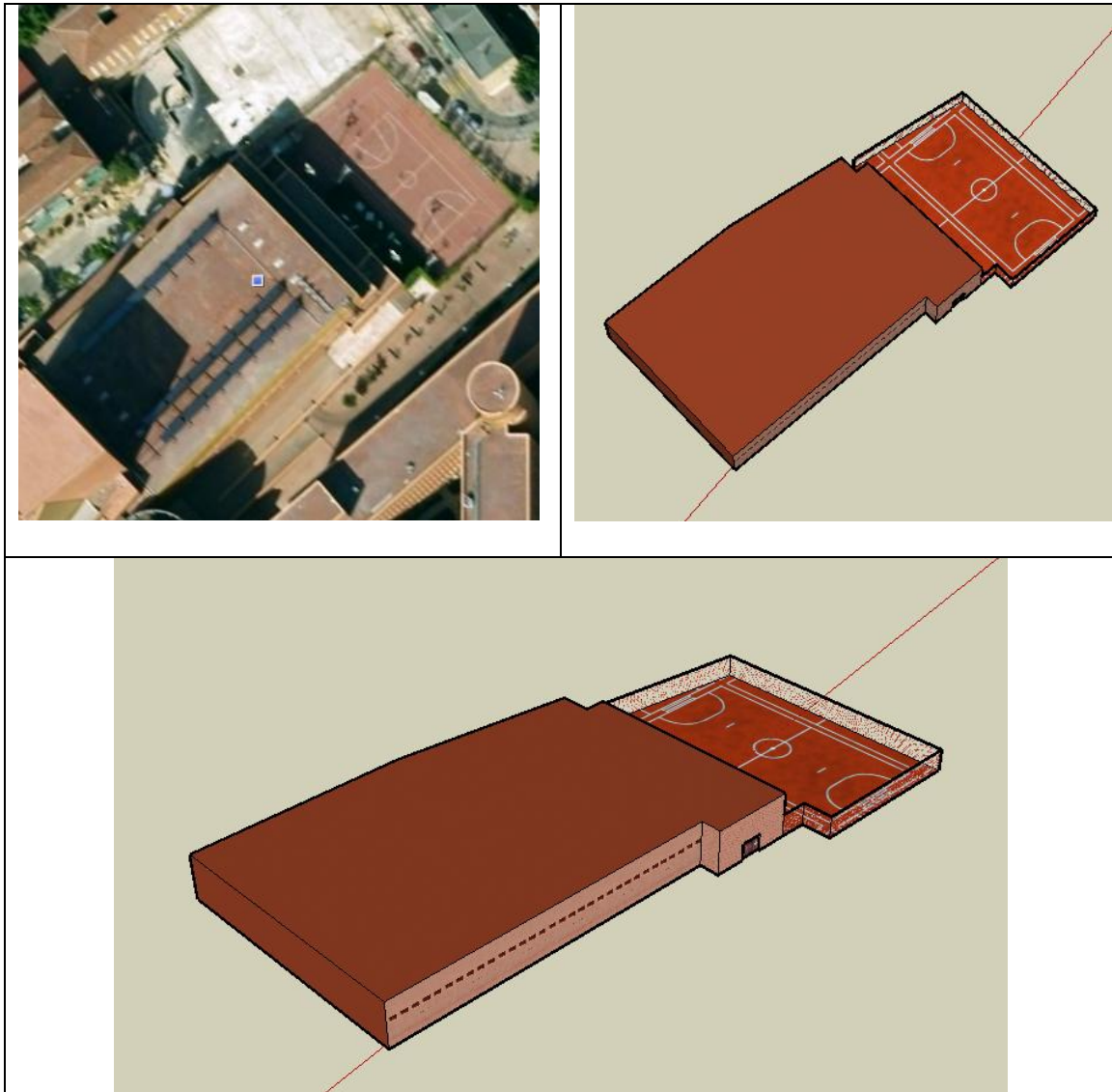
El edificio Juan Benet es el número 7 del campus de Leganes, y en su interior alberga lo siguiente:

- Departamentos Universitarios.
- Aulas.
- OTRI.



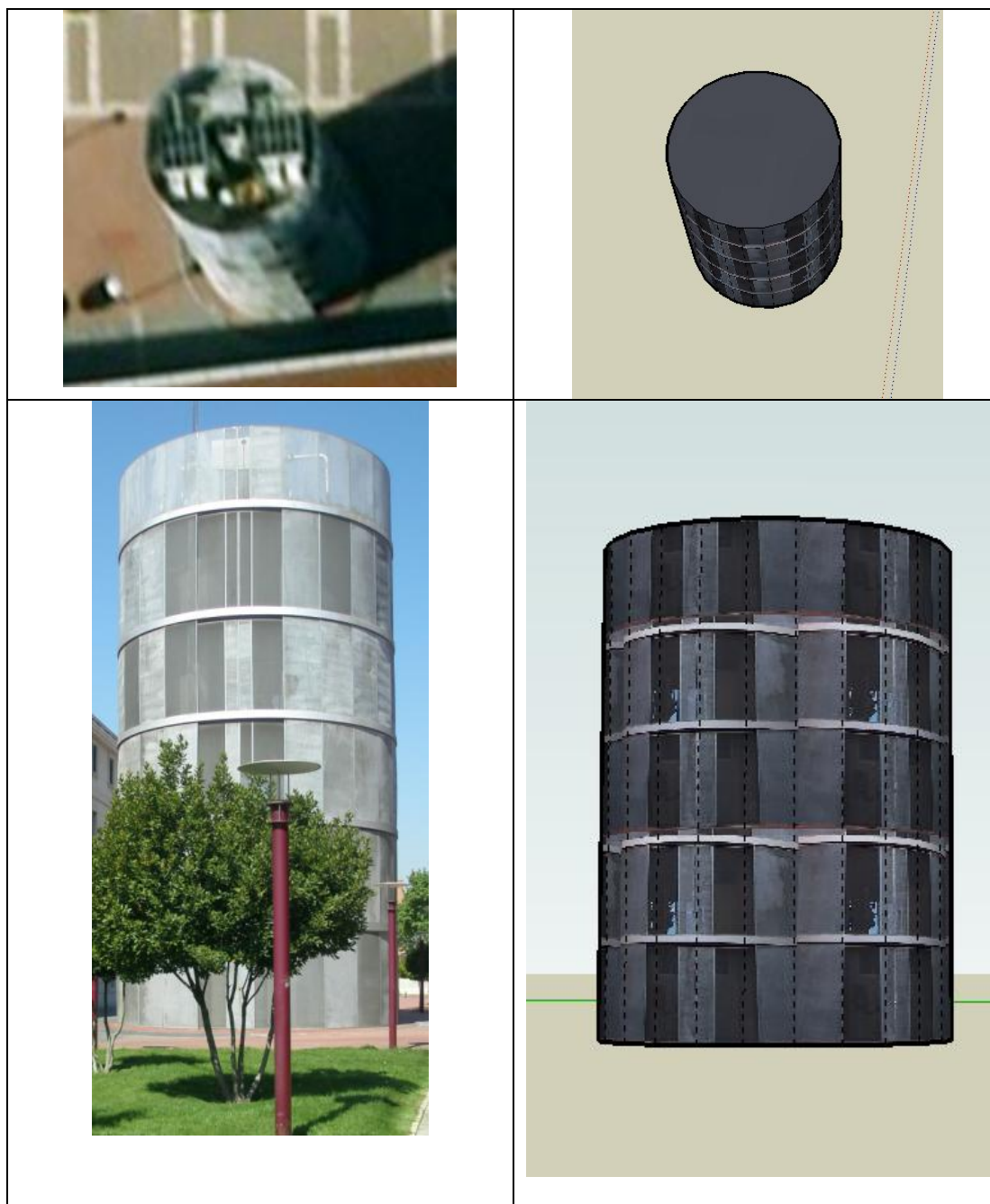
5.9. Polideportivo Alfredo Di Stefano.

El edificio Alfredo Di Stefano es el número 6 del campus de Leganes, y se utiliza como polideportivo.



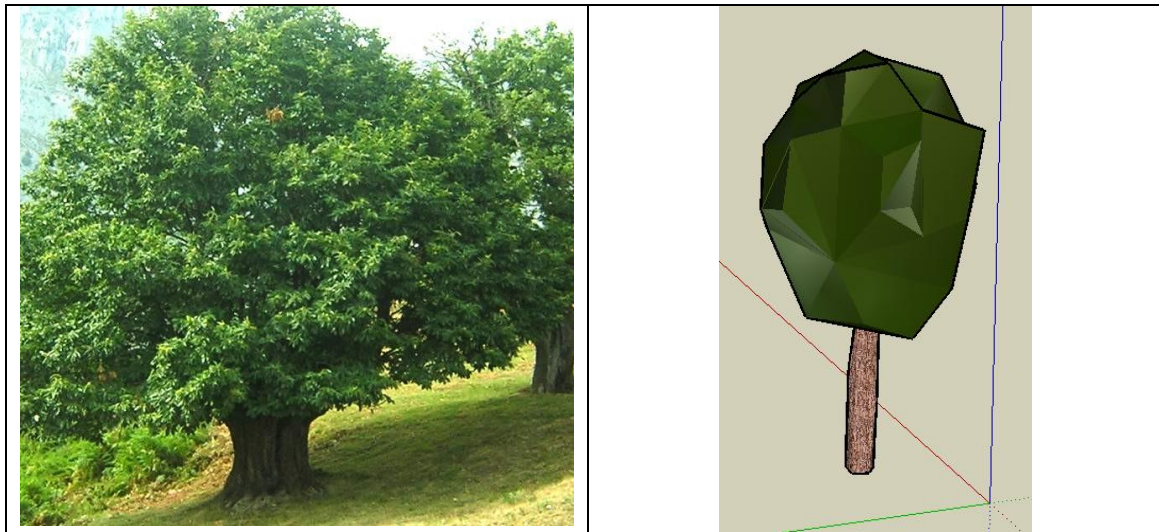
5.10. *Caldera.*

Situada entre el edificio Betancourt y Sabatini.



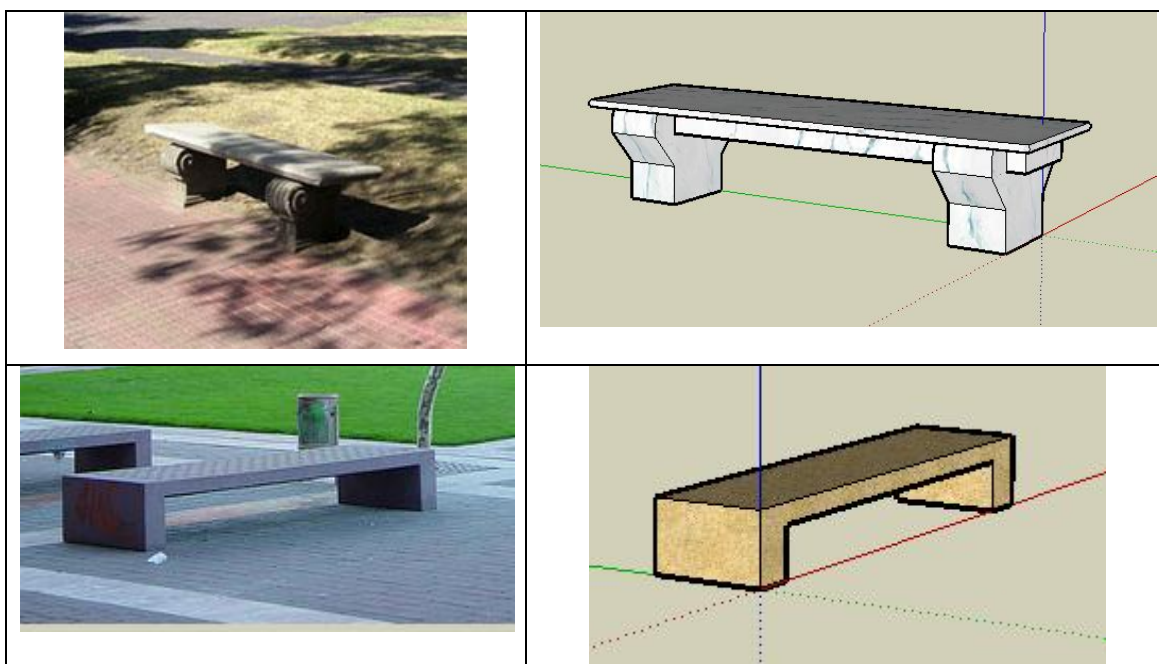
5.11. Vegetación

Esta será la única vegetación que podremos encontrar en el campus, aunque podrá variar la altura y grosor.



5.12. Bancos.

Dependiendo de la zona del campus que sea los bancos son de un tipo o de otro. Si nos situamos en la zona ajardinada del campus nos encontraremos el primer tipo, mientras que si nos situamos en la zona con baldosas, el formato será el segundo.



5.13. Señales de tráfico.

Las señales de tráfico verticales en España se basan en la normativa de la Unión Europea para que sean iguales en todos los países de la Unión (Convención de Viena).

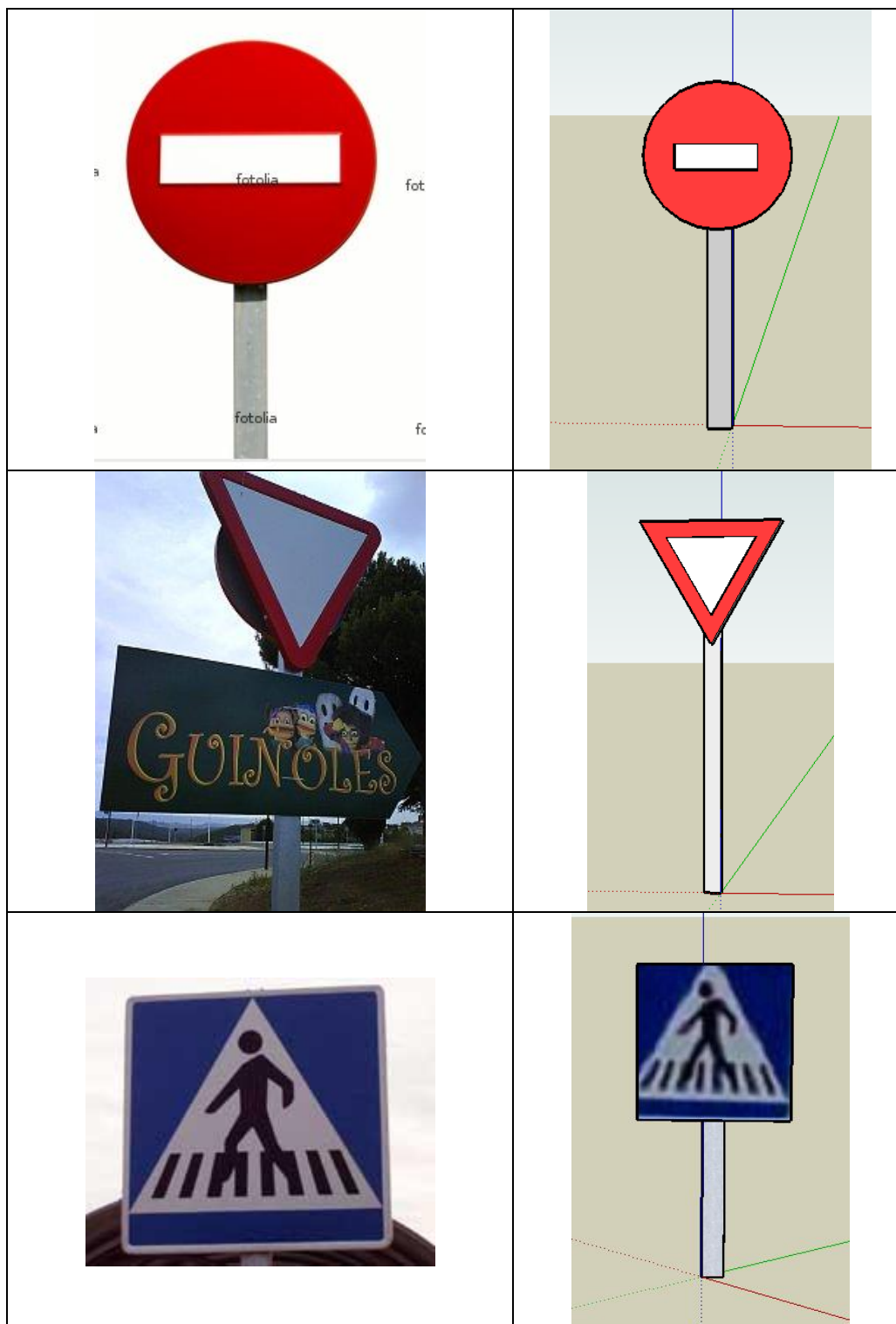
En nuestras opciones solo podremos seleccionar señales de tres tipos, los cuales son:

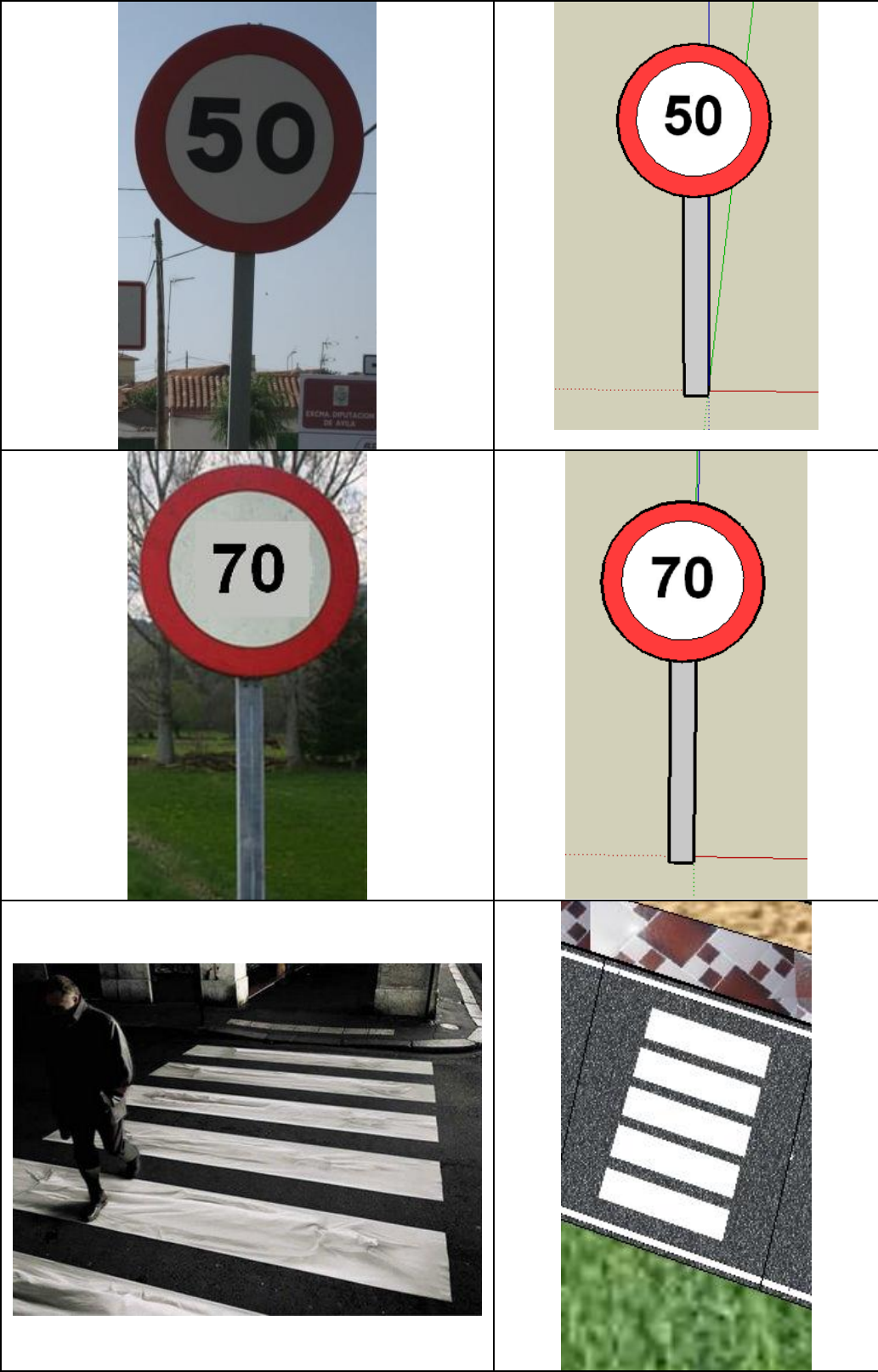
- De prioridad y prohibición de entrada.
- De prohibición.
- Indicaciones Generales.

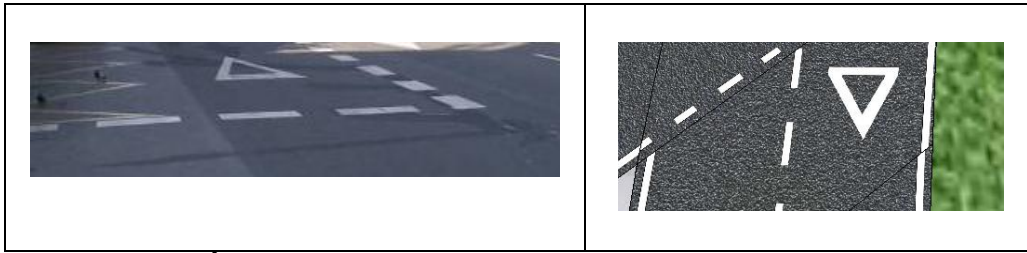
Según el grupo, las señales tienen formas y colores diferentes, con el fin de que puedan reconocerse con facilidad desde lejos:

1. Por su importancia, las señales de prioridad de paso tienen formas diferentes a todas las demás. La señal de "STOP" tiene forma octogonal y es de color rojo. La señal de prioridad de paso general, o "Ceda el paso", tiene forma de triángulo invertido y es de color blanco con borde rojo.
2. Las señales de prohibición tienen forma redonda y son blancas con borde rojo.
3. Las señales de final de prohibición tienen forma redonda y son blancas con tres líneas negras en diagonal.
4. Las señales de advertencia tienen forma de triángulo y son blancas con borde rojo.
5. Las señales de circulación tienen forma redonda o cuadrada, y son azules.
Las redondas indican obligatoriedad y las cuadradas son informativas.
6. Las señales de estacionamiento son redondas de color azul con borde rojo y una o dos líneas rojas en diagonal, o bien cuadradas o rectangulares de color azul. Las redondas indican prohibición y las cuadradas o rectangulares son informativas.
7. Las señales informativas tienen forma rectangular y son azules.

El uso de señales de tráfico en Europa viene de antiguo. El motivo es que las señales son comprensibles para los conductores de cualquier país, ya que no contienen texto. De esta forma los automovilistas europeos que viajaban, y viajan en la actualidad, por otro país en el que se habla un idioma distinto, no tienen dificultad en interpretar las indicaciones en las vías de circulación.







6. EVALUACIÓN Y PRUEBAS.

Vamos a realizar dos tipos de comprobaciones, por un lado, se verificarán los modelos 3D generados mediante Sketchup, será necesario que mantengan sus propiedades físicas, que su malla sea correcta, sus texturas,... es decir, que sea importado satisfactoriamente por MRDS, y por otro, se verificará el correcto funcionamiento del software de modificación

El resultado ha sido correcto en todas las pruebas, por lo que sólo expondremos las pruebas a las que los hemos sometido.

6.1. *Pruebas de los modelos 3D.*

Estas pruebas se aplicarán a cada uno de los modelos.

- Prueba 1 -> Exportación correcta de los .obj:

Al terminar de modelar un objeto mediante Sketchup, este te lo guarda automáticamente en un formato propio (.skp), que no reconoce Microsoft Robotics Studio, por lo que tendremos que exportarlo a .obj (formato que si reconoce) y comprobar que en esta exportación no se haya modificado o perdido nada del modelo.

- Prueba 2 -> Texturas mapeadas correctamente:

Se comprobará que al cargar un modelo en Microsoft Robotics Studio, las imágenes mapeadas se muestren correctamente.

- Prueba 3 -> Física correcta:

Se comprobará que al cargar un modelo en Microsoft Robotics Studio, la física del modelo se corresponda perfectamente con la realidad.

6.2. *Pruebas del software.*



- Prueba 1 -> Lectura correcta del fichero principal:

Al iniciar la aplicación se deberá cargar un fichero, que contiene el campus completo, con todos los modelos. Ya que a partir de este se genera el campus virtual deseado por el usuario, se comprobará que el software lee correctamente el fichero.

- Prueba 2 -> Navegación correcta por la interfaz:

Se comprobará que no hay ningún problema en la navegación por las distintas pestañas de la aplicación.

- Prueba 3 -> Generación correcta de diversos entornos virtuales:

Al terminar de seleccionar los modelos que queremos que formen parte de nuestro campus virtual, al guardarlo, generará un fichero que comprobaremos en el Visual Simulation Enviroment de Microsoft Robotics Developer Studio.

7. CONCLUSIONES Y FUTUROS DESARROLLOS.

7.1. *Ampliaciones futuras:*

Este entorno virtual está diseñado en un principio para realizar pruebas a un proyecto de investigación centrado en técnicas de fusión sensorial, donde la plataforma de experimentación consiste en un carrito de golf equipado con una serie de sensores, cada uno de los cuales estará encargado de recopilar un determinado tipo de información. Esta a su vez enviara a una unidad central, la cual se encargará de procesar toda esta información y a partir de ella desarrollar sistemas de asistencia a la conducción.

Debido a que la plataforma no es muy grande y no alcanza grandes velocidades, este entorno es perfecto, pero se puede aplicar a otro tipo de plataformas móviles (véase coches, aviones,..), para lo que se necesitaría un entorno mucho mayor, con lo que se podrían añadir nuevos modelos y más grandes, con lo que se podría generar pequeñas ciudades virtuales, con suficiente espacio para poder realizar las pruebas.

También si se desarrollasen nuevos algoritmos de detección visual por medio de cámaras de otro tipo de objetos (en este proyecto se prueba la detección de señales y personas) se podrían modelar e introducir tranquilamente en los entornos virtuales.

Como vemos sería muy sencillo poder ampliar este proyecto para que fuese capaz de generar cualquier tipo de entorno virtual.

7.2. Conclusiones.

Se puede decir que Sketchup es una herramienta de diseño 3D potente y de muy fácil manejo, por lo tanto recomendar esta herramienta no solo a usuarios con poca experiencia en el diseño 3D, sino también a los usuarios experimentados que necesitan desarrollar modelos 3D de forma rápida, pero con buenos resultados.

Otra conclusión a la que he llegado realizando este proyecto, la importancia de los entornos virtuales como entornos de pruebas, ya que pueden llegar a ahorrar grandes cantidades de dinero, al no tener realizar dichas pruebas. Por un error se podría perder el prototipo.

A continuación voy a hacer una breve descripción del proyecto **ASISTENTUR**, para demostrar el valor añadido que le da nuestro proyecto:

Este proyecto se centra en los sistemas de seguridad activa cuyo objetivo es dotar a los vehículos de sistemas inteligentes que predigan y eviten accidentes que el conductor por sí solo no puede controlar. De entre los diversos Sistemas Avanzados de Asistencia a la Conducción (ADAS) posibles, se pretende trabajar dentro de entornos urbanos. El sistema propuesto evaluará las maniobras indicando la posibilidad de que se estuviera cometiendo una imprudencia o bien informando sobre posibles peligros. Para ello se realizarán dos módulos de percepción básicos: detección de peatones y reconocimiento de señales de tráfico. Éstos se unirán al módulo de localización del vehículo en el módulo de fusión y razonamiento que será el encargado de evaluar los posibles peligros y avisar al conductor a través de una interfase amigable.

Módulo de Reconocimiento de Señales de Tráfico

La percepción de señales de tráfico es útil para comprobar la velocidad a la que se desplaza el vehículo, así como estar prevenidos ante posibles peligros que puedan aparecer. Su localización nos es fácil ya que existen otros objetos con el mismo color, pueden estar ocluidas y el sistema debe de funcionar ante condiciones luminosas muy diversas.

Para su detección se ha utilizado un modelo geométrico que tiene en cuenta además de la forma de la señal su color. Se ha llevado a cabo un estudio del color de las señales ante cambios debidos a la iluminación y a la edad de la señal. Para encontrar la señal presente en la imagen se han estudiado algoritmos de búsqueda basado en Algoritmos Genéticos (Genetic Algorithms) y Templado Simulado (Simulated Annealing).

El reconocimiento de la señales se complica al no estar totalmente estandarizado su diseño. Se ha estudiado el comportamiento de varias redes de neuronas frente a la correlación normalizada.



Aplicar Algoritmos de la IA a la identificación de señales de tráfico: módulo inteligente de señales de tráfico

Estado actual del proyecto:

- Se han abordado las tareas propuestas en el módulo inteligente de señales de tráfico, haciendo hincapié en los siguientes aspectos:
 - Algoritmo de crecimiento de regiones: se han detectado problemas por ser muy sensible el color rojo a cambios en la luminosidad.
 - Se han realizado trabajos y estudios para localizar y segmentar las señales, aún en las condiciones luminosas menos favorables.
 - Se han realizado estudios para encontrar una medida que permita conocer de forma automática si una señal es circular o no.
- Se ha trabajado en la reducción de atributos necesarios para identificar una señal de tráfico empleando conjuntos de clasificadores y otras técnicas subsimbólicas con el objetivo de reducir las dimensiones de las redes de neuronas artificiales empleadas.
 - los resultados obtenidos son alentadores, pese a obtenerse un porcentaje de aciertos un 4% más bajo en el caso más desfavorable. Se va a estudiar ahora la influencia dentro de una misma clase de señales.

Con esta breve descripción se puede observar la gran cantidad de valor añadido que aporta nuestro proyecto, al poder realizar todo tipo de pruebas en diferentes entornos diseñados exclusivamente para módulo de software de visión artificial. Estos entornos se crean de una manera muy sencilla y rápida sin la necesidad de tener conocimientos de modelado 3D o de la herramienta Microsoft Robotics Studio Developer.

También destacar la herramienta Microsoft Robotics Developer Studio, ya que es una herramienta potentísima donde se puede simular todo tipo de condiciones físicas, un punto en contra es la poca documentación que hay sobre esta herramienta. Para nuestras pruebas, es tremendamente útil, ya que podemos realizar los entornos virtuales a nuestro antojo, insertando modelos 3D, así como modificar la luz o la visibilidad, para ver como se comportan las cámaras y sensores con distintos tipos y cantidades de luz.

Para concluir se puede decir que este proyecto puede ser de gran utilidad para detectar los posibles fallos en los proyectos de investigación centrados en técnicas de fusión sensorial.



BIBLIOGRAFÍA.

- 1) **Sitio Web:** Sitio Web oficial de Sketchup.

URL: <http://sketchup.google.com/intl/es/>

- 2) **Sitio Web:** Sitio Web youtube, apartado de video tutoriales de Sketchup.

URL: <http://www.youtube.com/user/sketchupvideo?blend=3&ob=4>

- 3) **Sitio Web:** Sitio Web oficial de Microsoft Robotics Studio Developer.

URL: <http://msdn.microsoft.com/en-us/library/bb896715.aspx>

- 4) **Sitio Web:** Sitio Web conscious-robots el cual contiene un foro para resolver dudas sobre MRSD.

URL: <http://www.conscious-robots.com/es/foros-./microsoft-robotics-studio/showcat.html>

- 5) **Sitio Web:** Sitio Web oficial de Netbeans.

URL: <http://www.netbeans.org/>

- 6) **Sitio Web:** Sitio Web oficial de Wikipedia.

URL: <http://es.wikipedia.org/wiki/Wikipedia:Portada>

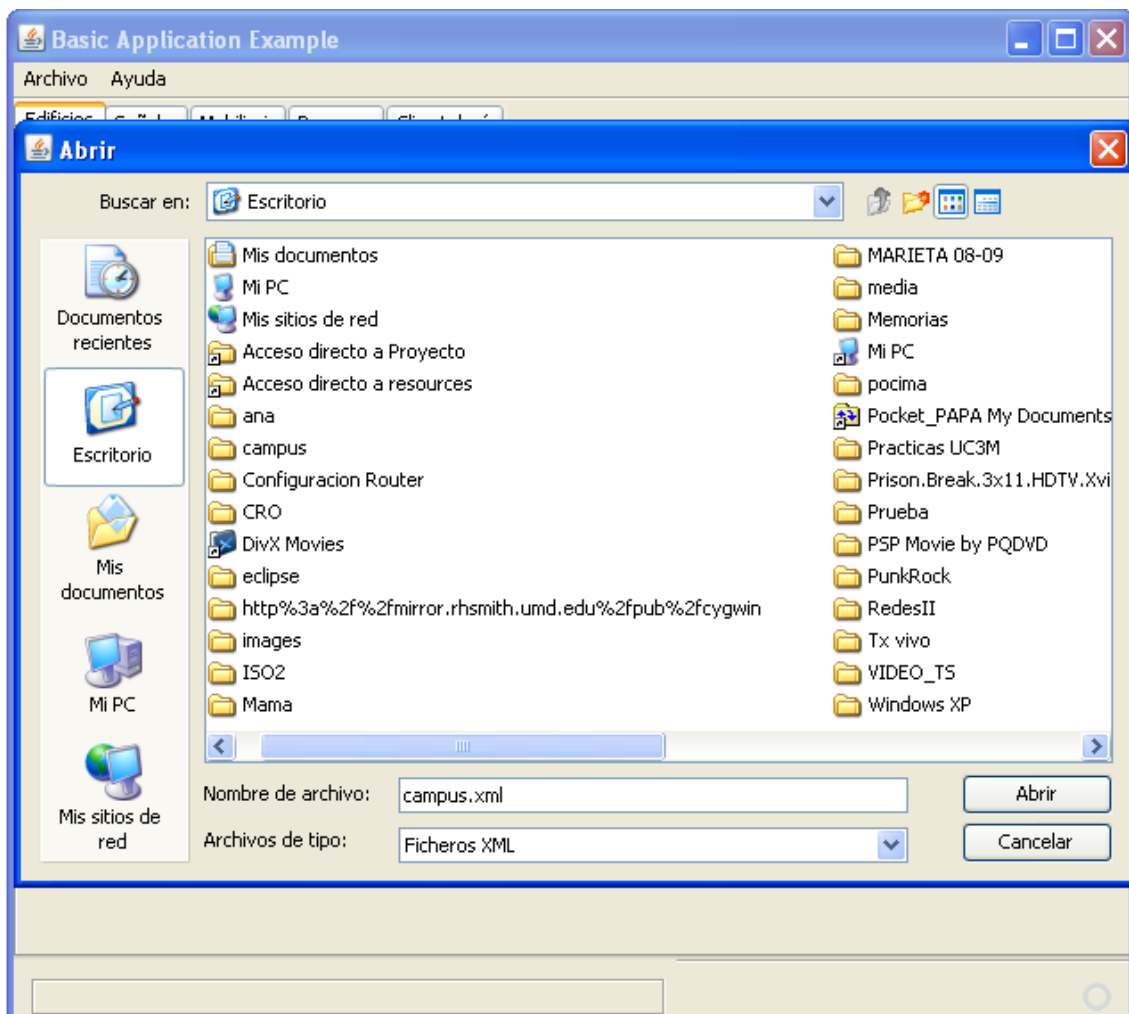
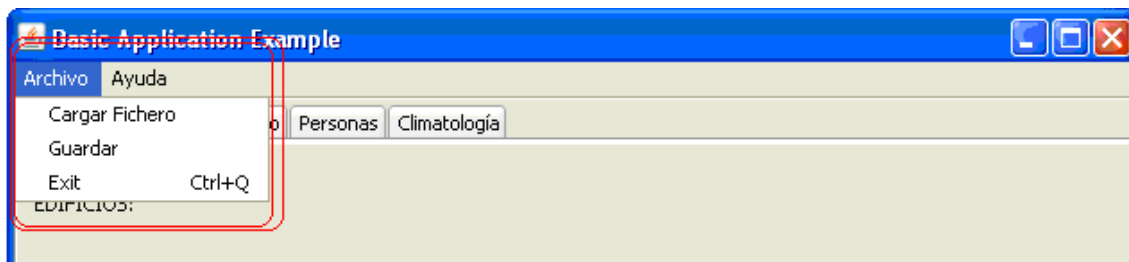
- 7) **Sitio Web:** Sitio Web oficial del proyecto ASISTENTUR.

URL: http://www.caos.inf.uc3m.es/index.php?option=com_content&task=view&id=21&Itemid=27

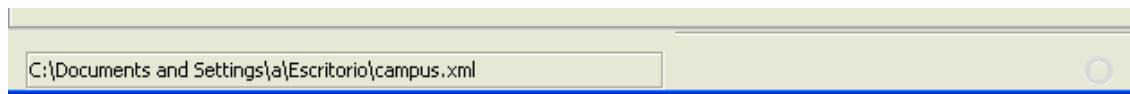
ANEXOS.

MANUAL DE USUARIOS:

Para comenzar se deberá cargar el fichero principal que se llama campus.xml y que se encuentra dentro de las carpetas de MRSD.

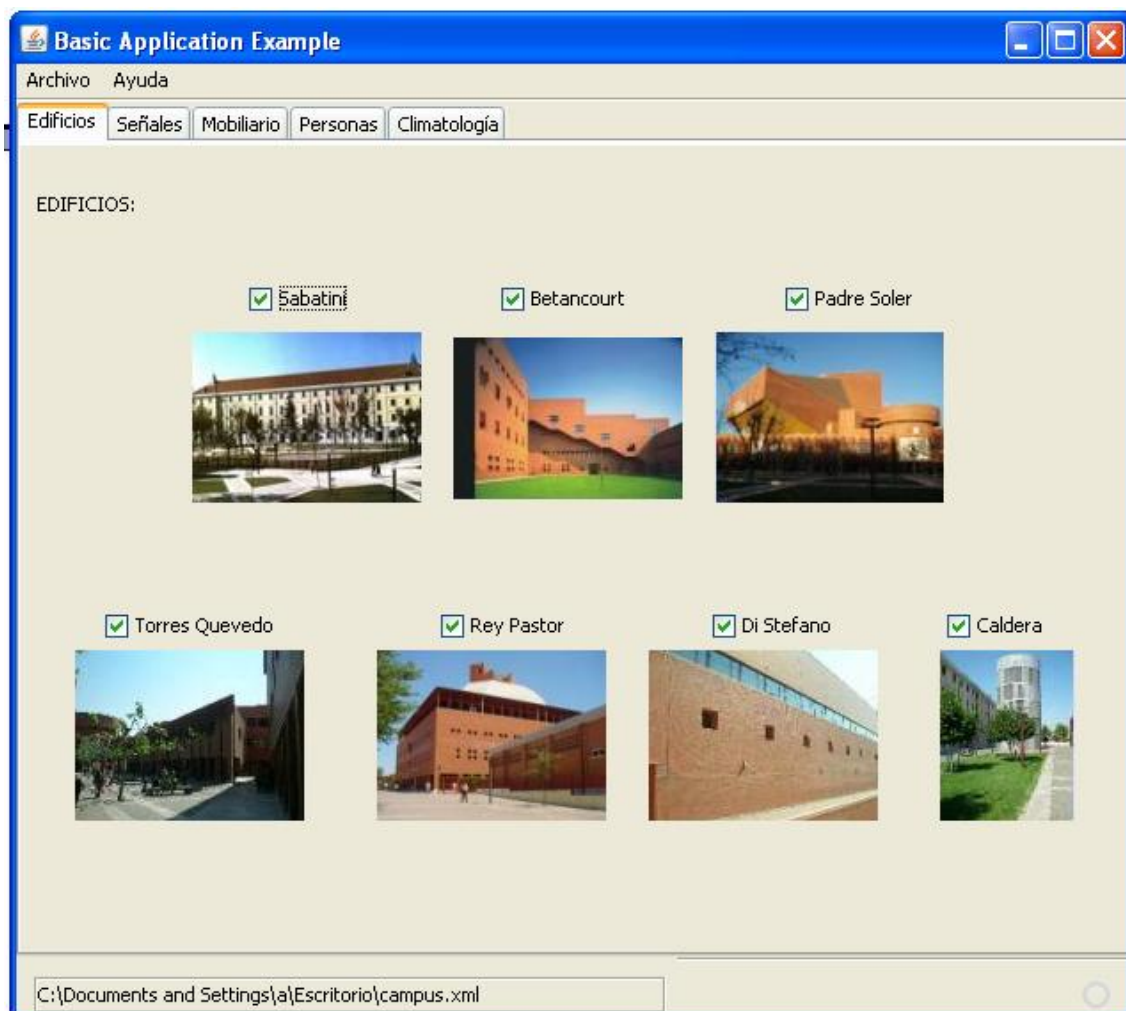


Comprobaremos que se ha cargado correctamente, cuando aparezca en la parte inferior izquierda de la interfaz la ruta del fichero.

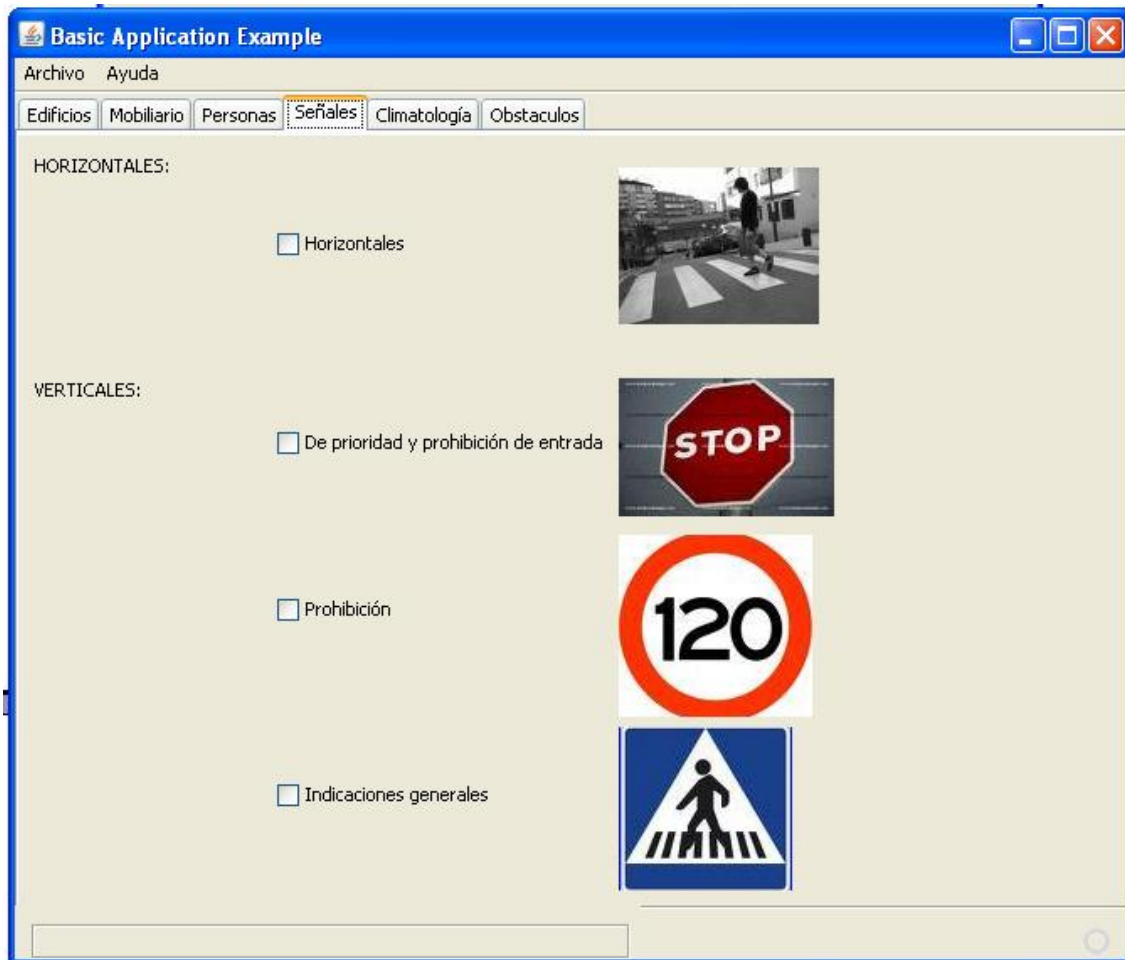


A continuación se seleccionarán los distintos modelos y opciones que queremos que formen nuestro campus, estos son:

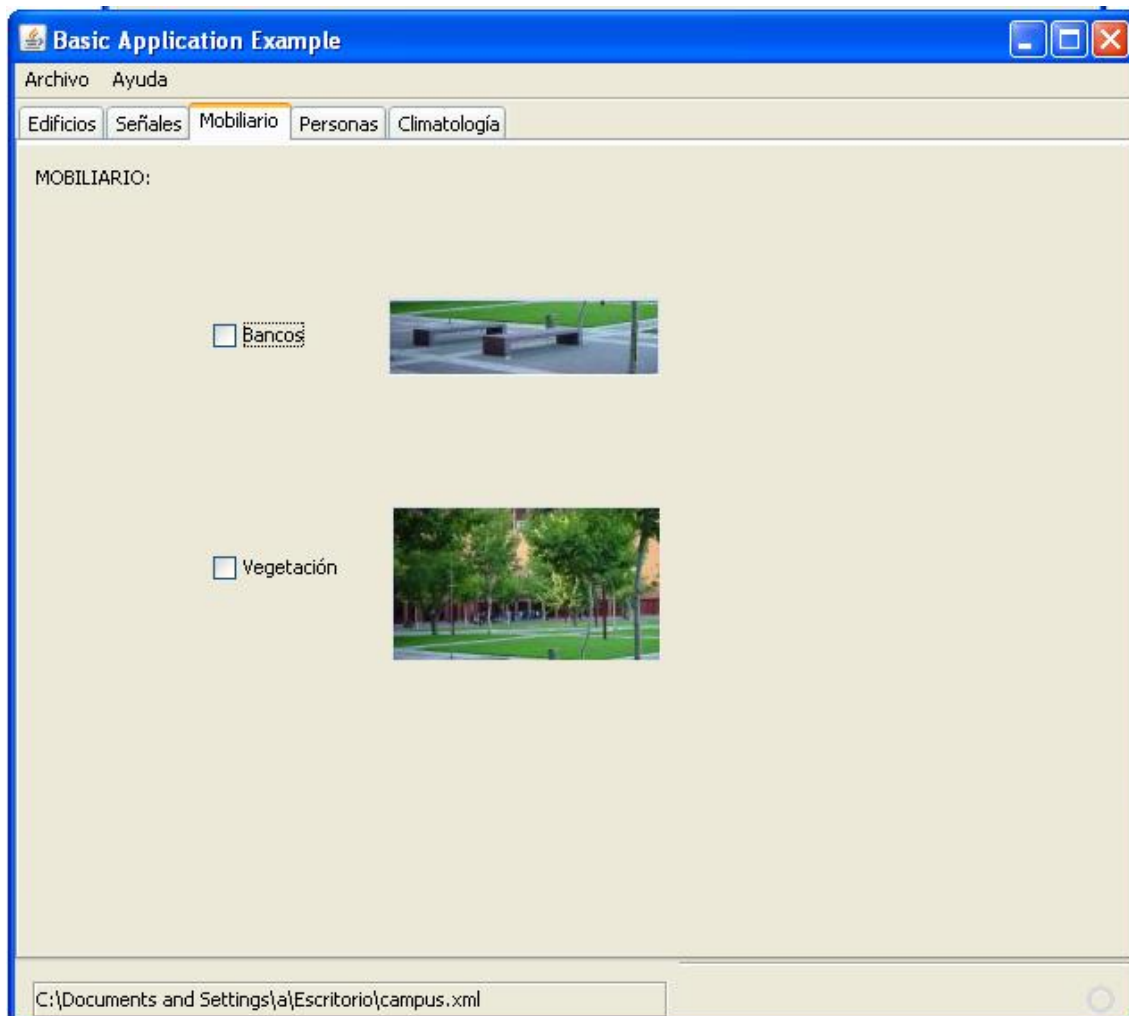
- Edición de Edificios: En este menú se podrán seleccionar los edificios que queremos que formen parte de nuestro campus virtual.



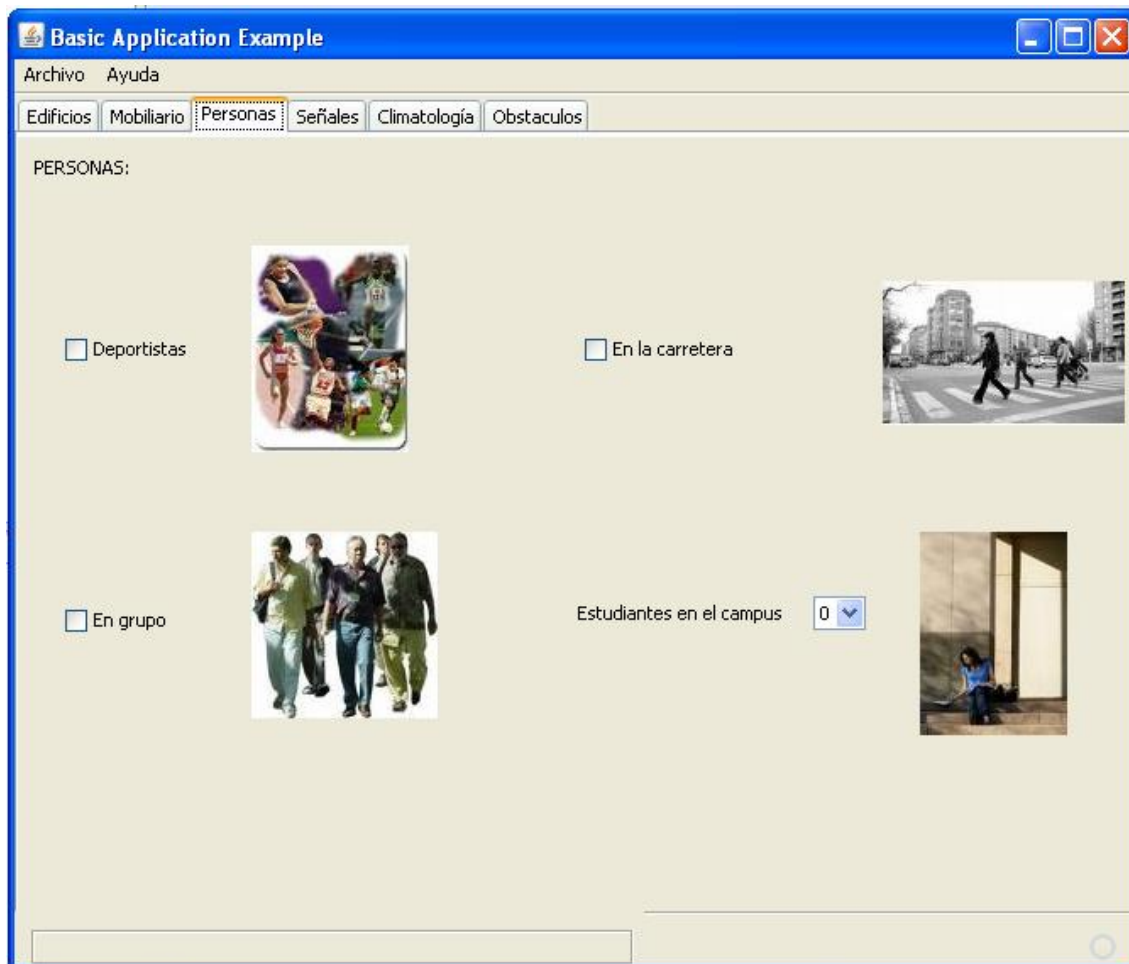
- Edición de Señales: En este menú se podrán seleccionar las señales que queremos que formen parte de nuestro campus virtual.



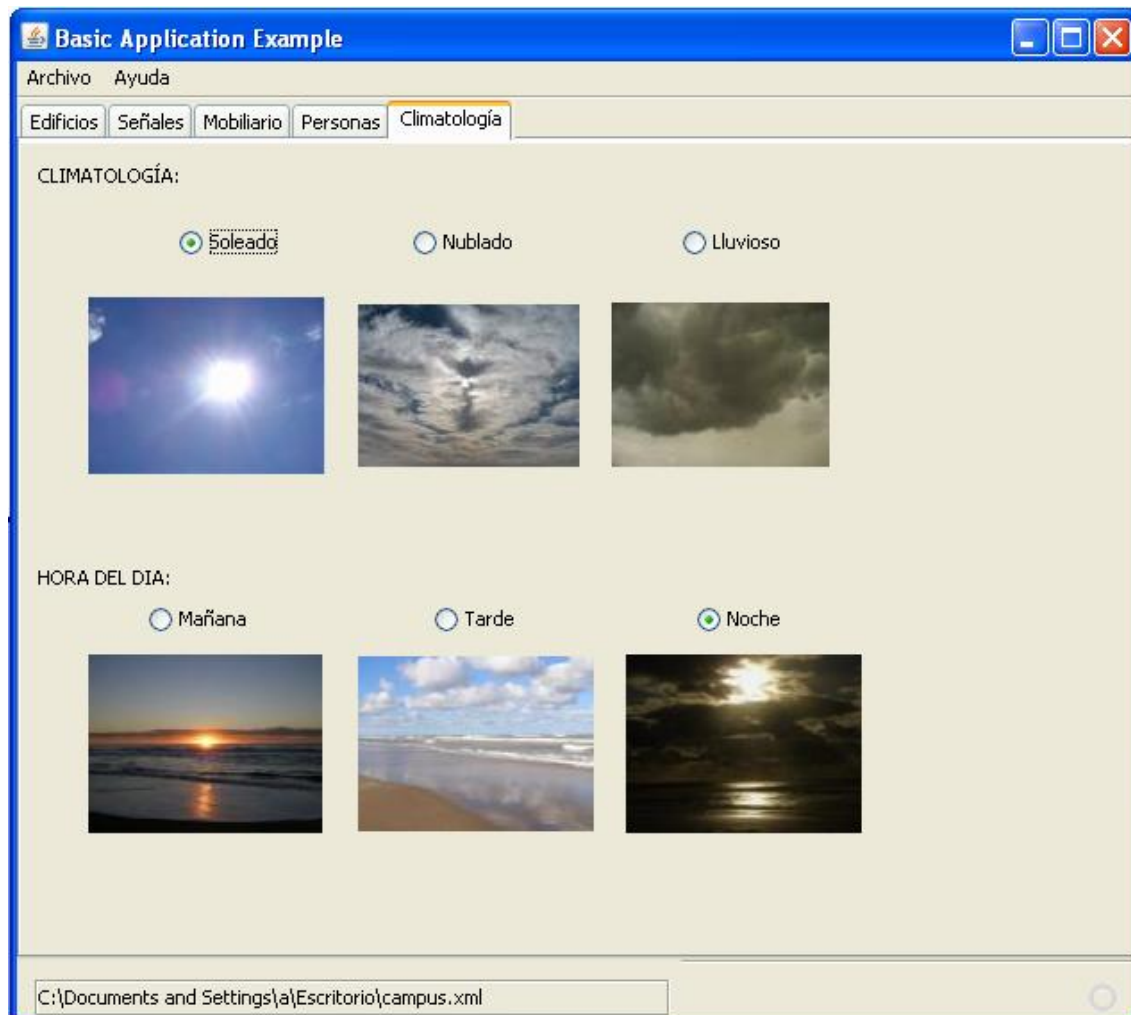
- Edición de Mobiliario: En este menú se podrá seleccionar si se desea que haya o no vegetación y bancos dentro de nuestro campus virtual.



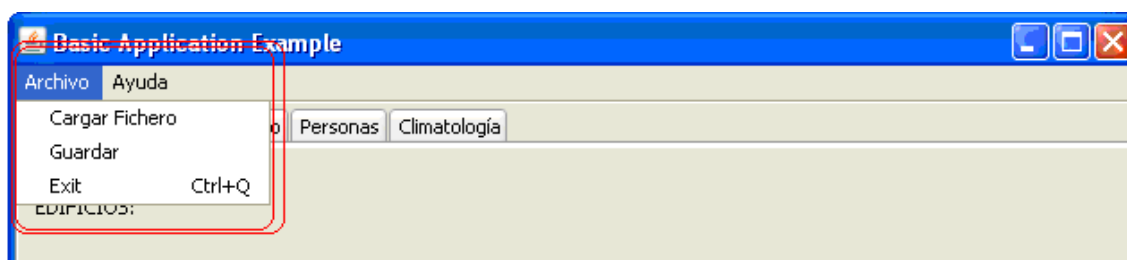
- Edición de Personas: En este menú se podrán seleccionar el tipo de personas que queremos que formen parte de nuestro campus virtual.



- Edición de Climatología: En este menú se podrá seleccionar el momento del día en el que el usuario quiera que se encuentre su entorno virtual y el tiempo que quiere que haga.



Una vez seleccionado todas las opciones se guardará el fichero en la misma carpeta donde se encuentra el fichero principal.



En cualquier momento se podrá visualizar este manual de usuario para cualquier tipo de ayuda, pulsando en “Manual Usuario”.

